Enhancing the Software Development Experience Using A Virtual Reality Integrated Development Environment

Dominik Ludera
40062500

Mike Calixte
40043608

Mina Selim
40043852

Abstract— **Human computer interaction has become one of the single most important aspects of emerging tools and technologies as we enter an age of technology that immerses our everyday. Software developers are often at the forefront of the development and usage of any and all legacy and new platforms. VR and AR have become the newest developments in a world seeking further immersion. AFrame and React are both tools that have been constructed for the further creation of these immersive environments. In this report, we present our research question on the potential increase in productivity in the development of software and the solving of complex problems using an integrated development environment due to largely to the immersive environment and increase in development "real-estate". "Real-Estate" refers to the amount of free visual space and pages that can be created within the 4th Wall IDE. The solution utilizes gestures that are picked up on by the VR headset as well as physical keyboard input. Furthermore, to validate our hypothesis as well as our proposed solution, we conducted a pre-trial and post-trial usage questionnaires with 6 participants. We have presented the result of this overall usability study including the end-user responses and further analysis. These end-users evaluated the layout and design of the platform as well as their productivity experienced by the users.**

I. INTRODUCTION

In an increasingly technology-driven and software immersed world, the process of software development has evolved significantly throughout in order to meet the demand and overall necessity for more software. Half a century ago, the software that got mankind to the moon was developed and written on paper. Nowadays, the process of software development has moved from the physical space to the digital space. IDEs (Integrated development environments) are one of the most important tools within the toolkit of developers. Good IDEs save a lot of development time and significantly increase the productivity of developers through the use of convenient tools, integrated capabilities and a tracking of the software life cycle itself.

In recent years, IDEs have gotten significantly complex, with multiple tabs and features. With the small screen real estate provided by the current computer screens, these features have a tendency to make development slower. The performance of developers is significantly impacted due to the decrease in usability and learnability of the newer IDEs.

A. Research Question

There is a surge of new technologies within the last decade, and one of the newer technologies that will take center front within this new decade is virtual reality. As code development moved from the physical space to the digital space, is the next logical transition to be from the digital space to the virtual space?

Virtual space contains near infinite (limited by processing power) amount of space. One important part of the transition between physical to digital was the increase in space (or screen real estate). According to various researches, an increase in screen real estate increases productivity, and we would like research if that can translate in a VR world for developers.

B. Hypothesis

We believe that a VR IDE will increase the productivity of the developers, permitting them to finish their tasks at a quicker rate compared to traditional IDEs thanks to the increase in screen real estate.

II. RELATED WORK

According to research done by the University of Utah[1] – "[...] more space means greater productivity until the point is reached where screen size and task requirements intersect". This hypothesis has been further supported by the results listed in their conclusion, a dual 20" screen setup has been shown a 44% increase in performance values over a single 18" monitor. Although dual screens have been preferred by the users, a single 24" widescreen has shown 6% performance over it. Following those results, and knowing from experience how developers' work spaces are cumbersome, we believe immersing users in a virtual space will allow them to better visualize different views they might use to complete their task and review it. There currently exist many applications that would allow a developer to develop in a VR environment. The simplest form of virtual development is through the usage of virtual desktops. Popular VR headsets such as the Oculus rift allows the users to put windows in the virtual spaces [2]. They would allow the developer to put multiple windows of his IDE on the multiple VR screens, as well as other supporting applications being run in different windows. This development environment however requires an expensive headset, and a powerful machine that can run the environment. It is not an affordable solution. That system is also not portable. Another existing commercial system is Primitive[3]. It allows an immersive development environment, with many visualization tools for the architecture of the code, file structure, etc. It works as a plugin for popular IDEs from Jetbrains and Microsoft. It is very powerful and useful, however, focuses only on code tree visualisation and isn't portable.

III. SYSTEM DESIGN

A. System Overview

In order to test our hypothesis, we created a VR IDE web app with the intention of having end-users solve algorithm or code freely. Once satisfied with their code, the user can compile and run the code right from the browser.
The application was built using a React and A-frame library for the front-end of the platform and a Java Springboot back-end for all code compilation. The front-end of the platform consists of the VR (and optional AR) integrated development environment visuals as well as the handlers for live interaction with the platform. The React and A-frame libraries were used to build an immersive virtual reality integrated development environment that will allow end-users to read and solve coding challenges. These challenges will be further expounded on in the evaluation and design sections. The React-Aframe NPM library used was version 4.4.0 made public 2 years ago and is currently still supported. The Springboot framework version was 2.2.1 and is also currently still supported. The platform itself was all built using the IntelliJ IDE as well as VSCode version 1.44 in order to create the back-end compilers and a React-AFrame front end VR environment respectively.

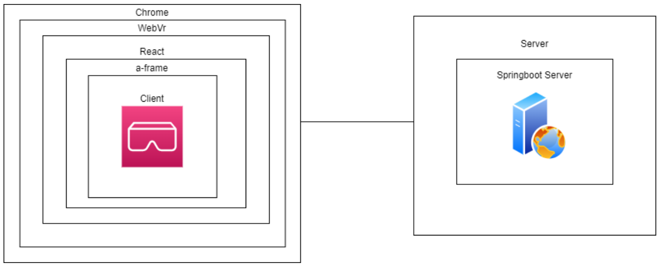


Fig. 1. 4th Wall platform architecture

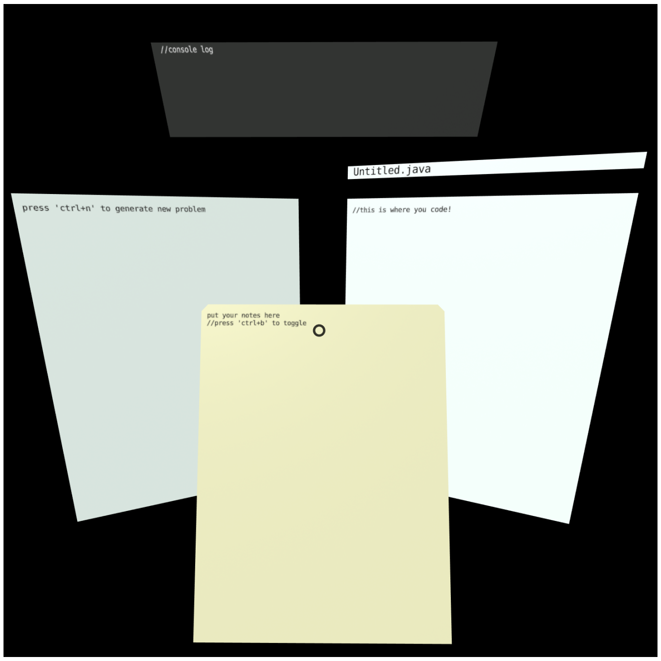
At a high level, the implementation of the app is as follows;

> The end-user takes on coding challenge and proceeds to submit their proposed solution to the challenge upon them determining it to be complete. Upon submission, a request to compile is created and sent to the back-end server. The request sent to a back-end server for compilation. When compilation of the code is complete, the back-end Springboot server returns a response containing the output of the code itself. The end-user can then determine if their proposed solution was correct or not.

Shortcuts are provided to the end-user to streamline the creation of a few new text spaces and pages in order to facilitate the overall development experience.

B. Feature Overview

The system's design makes use of conventional textarea frames which represent different windows with which the user can interact. As shown in *Fig. 2.*, basic functionality is



```
//console log
```

```
Untitled.java
```

```
press 'ctrl+n' to generate new problem
```

```
//this is where you code!
```

```
put your notes here
//press 'ctrl+b' to toggle
```

Fig. 2. 4th Wall platform usage model

provided to the user.

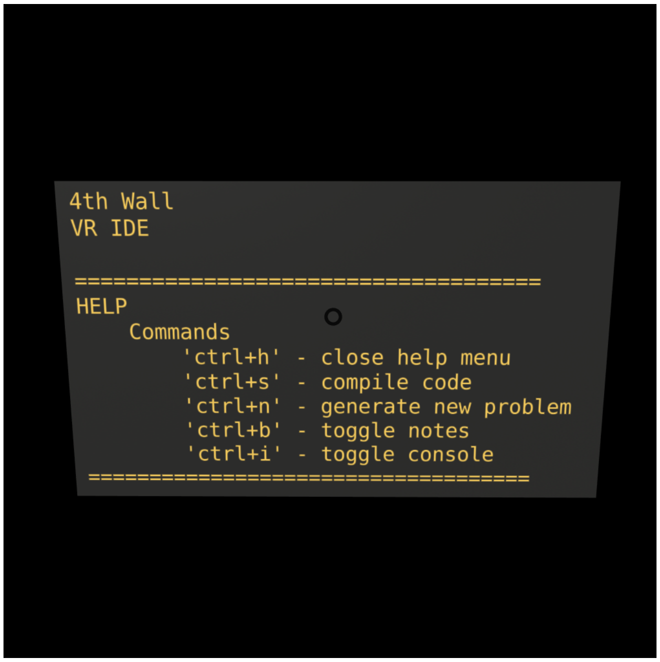
1) VR Space: The VR Space have been organized to provide a somewhat intuitive approach. The user is greeted with a help page, summarizing the different keyboard shortcuts necessary to operate the app.

In order to toggle the help menu off, the user has to press the appropriate keyboard shortcut listed under commands. This design was implemented in order to have an implicit feedback from the user, that it understands what is being displayed and how to use the keyboard shortcuts. The user can always go back and forth by toggling the help menu.

2) Problem Description Window: The left hand side provides the problem to be solved, it's goal is to emulate what a developer would have as documentation or problem definition. For the purpose of this study, pre-written problems have been provided, which the user is free to tackle. The user can generate new problems, by using the appropriate shortcut.

3) Notes Window: The Notes window serves as a draft sheet that developers would need to write pseudo code, or take notes appropriate to their task.

4) Coding Window: For this study, the coding window provides the user the ability to code JAVA. The window contains a title above it, where the user can name the file.
Note: The name of the file has to correspond to the class of the file.
When the user is done, it can compile and run the code, straight from the app and the result will be displayed in the



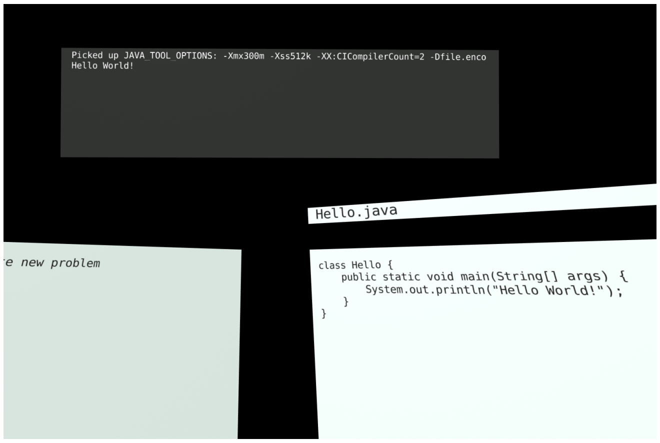
```
4th Wall
VR IDE

====================================
HELP
    Commands
        'ctrl+h' - close help menu
        'ctrl+s' - compile code
        'ctrl+n' - generate new problem
        'ctrl+b' - toggle notes
        'ctrl+i' - toggle console
====================================
```

Fig. 3. 4th Wall platform usage model

console window.

5) Console Window: The console logs the compilation results that the user types in the coding window. It is there to display syntax errors, runtime errors, compilation results or console outputs from the coding window.



```
Picked up JAVA_TOOL_OPTIONS: -Xmx300m -Xss512k -XX:CICompilerCount=2 -Dfile.enco
Hello World!
```

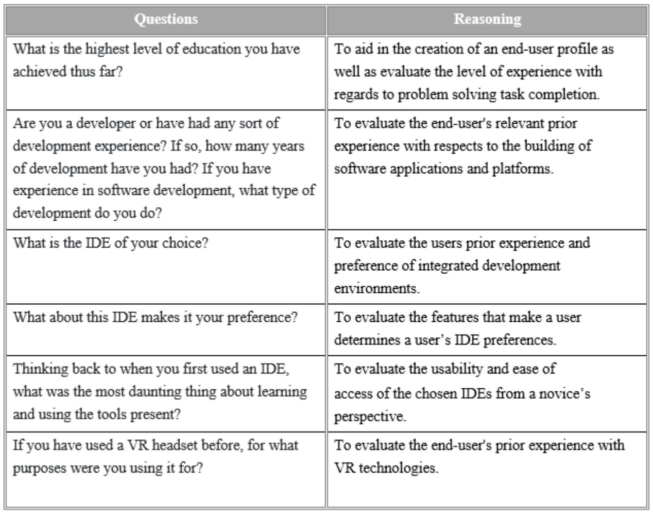
```
class Hello {
    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```

Fig. 4. 4th Wall platform usage model

IV. EVALUATION METHOD

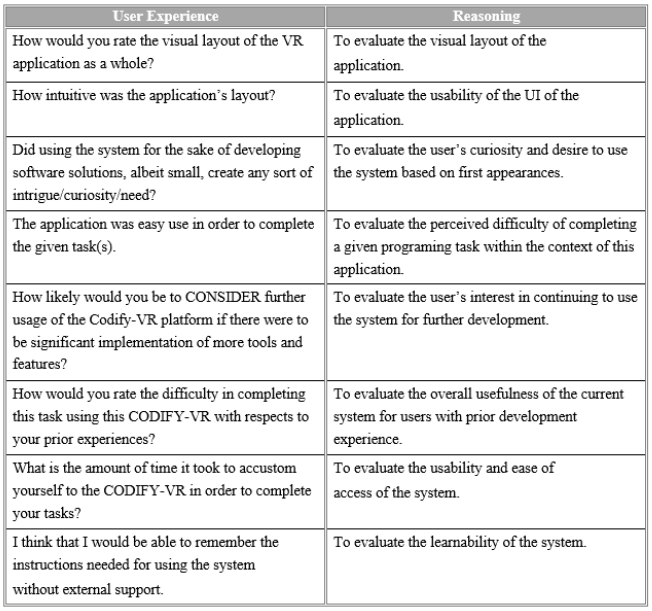
A. Study Design

The study design approach used throughout the study of this project and its subsequent evaluation makes use of the Think-Aloud approach. The approach will allow for the collection of the user's opinion whilst using the application as a novice or an end-user accustomed to the use of the application through multiple use or trial passes. As there exists a variety of different end-users varying in levels of development experience and more, it is necessary to ask preliminary questions in order to create general profiles and describe the backgrounds of our end-users. This will allow for the collection of various points of improvements and remarks as well as allow us to correlate these remarks with the various types of end-user profiles of this platform. This becomes quite significant since these user profiles will allow us to further envision and ultimately discuss end-user archetypes.



Questions	Reasoning
What is the highest level of education you have achieved thus far?	To aid in the creation of an end-user profile as well as evaluate the level of experience with regards to problem solving task completion.
Are you a developer or have had any sort of development experience? If so, how many years of development have you had? If you have experience in software development, what type of development do you do?	To evaluate the end-user's relevant prior experience with respects to the building of software applications and platforms.
What is the IDE of your choice?	To evaluate the users prior experience and preference of integrated development environments.
What about this IDE makes it your preference?	To evaluate the features that make a user determines a user's IDE preferences.
Thinking back to when you first used an IDE, what was the most daunting thing about learning and using the tools present?	To evaluate the usability and ease of access of the chosen IDEs from a novice's perspective.
If you have used a VR headset before, for what purposes were you using it for?	To evaluate the end-user's prior experience with VR technologies.

Fig. 5. Pre-Trial Questionnaire



User Experience	Reasoning
How would you rate the visual layout of the VR application as a whole?	To evaluate the visual layout of the application.
How intuitive was the application's layout?	To evaluate the usability of the UI of the application.
Did using the system for the sake of developing software solutions, albeit small, create any sort of intrigue/curiosity/need?	To evaluate the user's curiosity and desire to use the system based on first appearances.
The application was easy use in order to complete the given task(s).	To evaluate the perceived difficulty of completing a given programing task within the context of this application.
How likely would you be to CONSIDER further usage of the Codify-VR platform if there were to be significant implementation of more tools and features?	To evaluate the user's interest in continuing to use the system for further development.
How would you rate the difficulty in completing this task using this CODIFY-VR with respects to your prior experiences?	To evaluate the overall usefulness of the current system for users with prior development experience.
What is the amount of time it took to accustom yourself to the CODIFY-VR in order to complete your tasks?	To evaluate the usability and ease of access of the system.
I think that I would be able to remember the instructions needed for using the system without external support.	To evaluate the learnability of the system.

Fig. 6. Post-Trial Questionnaire

B. Adapted Study Design

Due to current, unforeseen circumstances, a video of the usage of the 4th Wall was also sent to users who were unable to run the current application using the online deployed version through their browser. This video will allow users to evaluate the 4th Wall platform, its usability and answer all of the required questions of the post-trial questionnaire. As for users who still have access to the required technology such as VR headsets or the capacity to run them on available mobile devices, the first-person experience and their subsequent evaluation of this experience will be accompanied by the post-trial questionnaire. It is to be noted that both users with and without the required tools necessary to run the application will still be asked to provide out load feedback which will be noted.

C. *Evaluation Methodology*

The study entails the performance of 3 problem solving challenges issued at random by the platform itself. These problems are approximately of the same difficulty. Difficulty is quantified by the amount of time taken to propose any correct solution to the problem. Each of the problems should take approximately 20 minutes to solve each.

On first load of the 4th Wall IDE, the user will go through a brief onboarding period which will provide the user with all necessary shortcuts to be used within the application. The user's interaction time for menial as well as the main task will be noted and questions concerning these times and the in-detail and overall experiences will be asked about in the post-trial survey as mentioned in the previous sections.

Our overall approach to the evaluation is actually two-fold – assess the user experience with the 4th Wall integrated development environment and to further compare users' experience with the first-time usage of a VR-IDE versus their own first-time usage experiences with an IDE or IDE-like tool. The provided answers to the post-trial survey combined with the feedback collected during remote-call testing of the 4th Wall IDE will provide us with what we believe to be sufficient information in order to evaluate the strengths and weaknesses of the current platform in comparison to the already available IDEs and coding tools in market.

The main focus of the 4th Wall platform has been learnability, user experience and general productivity while using the platform itself. Therefore, the questions asked during the pre-trial and post-trial surveys will allow for the collection of data that will reflect and evaluate these criteria.

INSTRUCTIONS FOR THE USERS

- Open VR compatible browswer (on mobile or VR headset).
- Connect bluetooth keyboard.
- Go to following link https://vr-ide.herokuapp.com/
- Read the first prompt.
- Start using the app.

D. *Study Recruitment Plan*

During our initial assessment and creation of the recruitment plan, it was decided that we would find an equal amount of experienced developers who were knowledgeable about IDE and IDE tools and novice developers who had little to no experience.

Unfortunately, due to unforeseen circumstances, it became only possible to gather developers who have had experience. Fortunately, most metrics and the study design are still very much applicable. All participants had not seen or used the 4th Wall IDE whatsoever before the trials and were unbiased in their judgment and feedback with respects to any internal improvement that may have occurred during the development time of the platform.

V. RESULTS

Given that the platform revolves around an immersive VR environment, the user interface and layout of the application or extremely important. The user interface plays the role as the main stage where all activity and platform interactions will occur. Furthermore, a comprehensive and enjoyable user interface also is a main metric in any users' evaluation of their satisfaction with respects to many if not all software driven platforms.

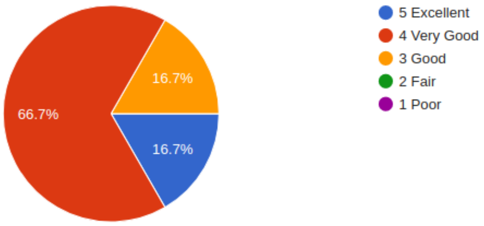


Fig. 7. Layout Evaluation Results

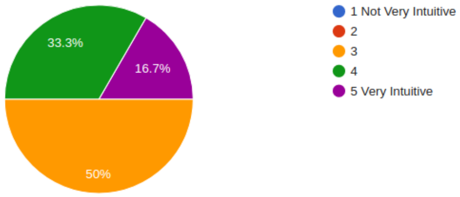


Fig. 8. Intuitive Design Evaluation Results

VI. DISCUSSION

A. *Interpretation of the Results*

From the data in figure A, we realize that 66.7% of all users found the platform's layout to be very good, while others rated the platform's layout as being either good or excellent. This tells us that the 4th Wall platform provides an engaging layout that many users find appealing. In figure B, we see that half of the users find the platform to be at worst neutral in terms of finding the user interface to be intuitive. 33.3% found it to be either intuitive or very intuitive. This distribution of answers was found to be due to 2 main reasons.

The first reason being that the application was forced to be loaded and deployed onto browsers. This often led to a discrepancy between the shortcuts that were meant to be used by end-users and the shortcuts available to be used on browsers from a device that is not a VR headset and Bluetooth keyboard combination.

The second reason being the use of the application through devices and/or the visualization of the usage of the platform through video. The former presents an obvious challenge to end-users in terms of usability and the latter forces "end-users" to become spectators instead of actual users of the final product. This third-person view leaves spectators attempting to understand the actual end-users' actions and not experiencing, evaluating and using the platform themselves.

Despite the challenges experienced by end-users, a large portion seemed to still have believed these challenges to be enough to hinder their ability to understand the purposes, methods of utility and layout of the application and results tended to be favorable.

Concerning the learnability of the application, the vast majority of users found the application's onboarding to be useful. Furthermore, most would somewhat agree or fully agree that given the chance to use the application again, they would fully remember how to use the application with little to no difficulty.

However, despite these promising results, not all feedback was positive. The vast majority of our end-users who actually got to use the application first-hand, found that the platform required trial and error before fully understanding how to use the platform successfully. Furthermore, only half off the users only somewhat agree that the onboarding process was fully clear. This may be due to the onboarding screen resembling console instructions instead of clear and concise interactive instructions as to how to use the platform correctly. These weak points of the platform will be further listed and described in the next section.

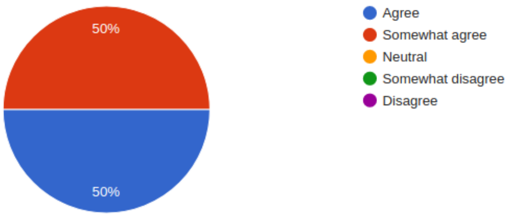


Fig. 9. Ease Of Task Completion

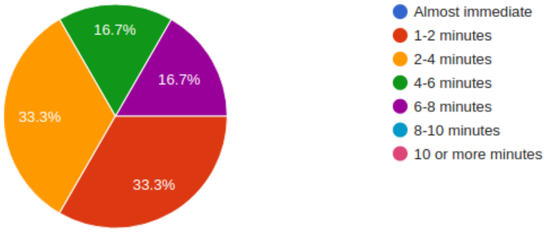


Fig. 10. Time Needed for complete understanding of the platform

Figure1 shows that all users either agree or somewhat agree to the face that it is easy to complete the coding tasks presented to them. Furthermore, according to *Figure2*, the vast majority of users only required 1 to 4 minutes to fully grasp the usage of the 4th Wall IDE platform. At the most, it took 6-8 minutes in order for some users to grasp the tools built into the VR IDE. These promising results may point to further in field usage capabilities for this platform.

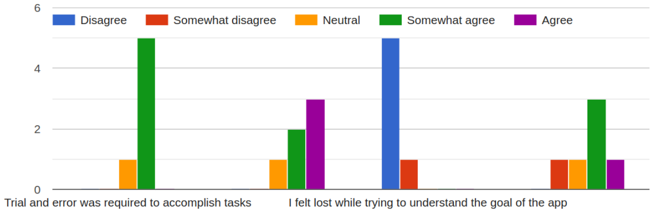


Fig. 11. Learnability Results and Opinions

Despite other promising preliminary results concerning the learnability and intuitive usage of the platform, it is clear that the experienced developers find this platform to either be slightly more difficult to use or the exact same in terms of usability with respects to the IDEs that they currently use. This, however is to be expected. Most of the developers who have used this application have had 3 or more years of time to develop tool, language and workflow preferences. Therefore, the usage of an entirely new platform being deemed as more difficult is reasonable.

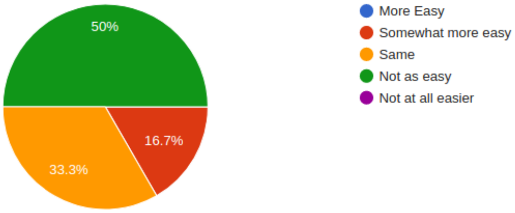


Fig. 12. Relative Difficulty with respects to other IDEs

When asked as to why these experienced developers deemed the platform more difficult to use when compared to other IDEs, the common reason cited was a lack of auto suggestions, auto-completions and refactoring tools. Therefore, in order to gauge future interest in the availability of such a tool on the market, the following question was

asked: “If the integration of more tools within the 4th Wall IDE such as code completion, auto suggestions, and more were to occur, would you consider using 4th Wall as one of the main IDEs that you use? Why or why not?”.
Most answers to this question were “yes”. Further quoted answers are listed in the appendix section.

B. Limitations of 4th Wall

A list of the weak points is listed below and are based on the opinions of the end-users, video spectators and our own opinions:

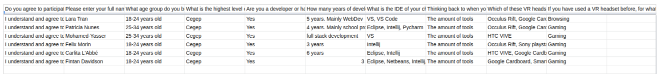
- The platform does not provide enough developer-oriented tools such as autocompletions, autosuggestions and refactoring tools.
- The platform does not provide a clear enough onboarding procedure for tips, tricks and shortcuts that are not and would not be intuitively grasped by a first-time user of the platform.
- The platform requires the usage of a Bluetooth or connected lightweight keyboard. For users not adept to typing or inputting information into some type of medium, this presents itself to be a serious inconvenience and challenge. Since a virtual keyboard is not presented to the user, this creates a case of trial and error when it comes to writing syntax error free code. Therefore, the input directives need to be significantly improved.
- The use of the a-frame software library, which is currently a library that has not had any significant release or improvement since 2018. Therefore, a move to unity may be more appropriate.
- The impossibility of interacting with the real world. Many users actually deem the VR very immersive, but were of the opinion that an augmented reality version would present even more significant user experience improvement. The AR version of the 4th Wall IDE is available, however the input methodology has yet to be implemented as and AR application was not the envisioned end product.

VII. Conclusion

The aim of our project was to create an immersive VR environment that would allow developers, both experienced and inexperienced, to be productive and able to complete their everyday tasks using an alternative more portable IDE. In order to measure if this platform was a viable alternative, we studied the usability, learnability of the user experience as well as the effort exerted by out end-users to successfully solve complex problems.
In order to do this, we implemented the 4th Wall IDE using a-frame and react technologies as well as a deployed compiler that is written in the Java programming language. To further aid in the evaluation of our hypothesis, we also performed a usability test where a set of tasks were prescribed. We then conducted a subjective comparison by collecting feedback from end-users.
From our study, we determined that the 4th Wall IDE as it is, is a viable option for solving complex coding problems but lacks enough integrated tools to be deemed market ready for more seasoned developers who have their own tool preferences. We also determined that there is serious intrigue in having this option to develop software a more immersive AR or VR platform. Since most of our users have some experience in the usage and creation of VR and AR application, it is reasonable to say that the feedback concerning the relative difficulty is more due to a lack of maturity in the overall development of the IDE. Given the proof of concept nature of the current 4th Wall version, this was deemed as reasonable.
However, it can be concluded that our hypothesis does not entirely hold true. Though it is true that our platform requires very little onboarding time and contains little challenge to understand and use in comparison to more mature IDEs, it is clear that most seasoned developers did not see a substantial increase in productivity for solving the same type of problems on regular IDEs. Despite this, we can say that there are also obvious technical benefits to the end-users as they no longer need to be mindful of resource intensive activities on VR headsets becoming too cumbersome for the hardware to handle. Overall, there is a scope for improvement of the 4th Wall IDE platform in terms of provision of tools for further productivity and more intuitive, concise and clear onboarding despite the platform itself being intuitive for most end users. These improvements could lead to a platform that could revolutionize the way developers solve complex problems on the go and at work.

VIII. Appendix

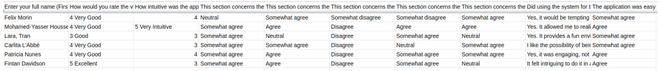
A. Appendix 1: Raw Data - Pre-Test Questionnaire



Do you agree to participat	Please enter your full nan	What age group do you b	What is the highest level	Are you a developer or ha	How many years of devel	What is the IDE of your cl	Thinking back to when yo	Which of these VR heads	If you have used a VR headset before, for what	
I understand and agree tc	Lara Tran	18-24 years old	Cegep	Yes	5 years. Mainly WebDev	VS, VS Code	The amount of tools	Occulus Rift, Google Carc	Browsing	
I understand and agree tc	Patricia Nunes	25-34 years old	Cegep	Yes	4 years. Mainly school prc	Eclipse, Intellij, Pycharm	The amount of tools	Occulus Rift, Google Carc	Gaming	
I understand and agree tc	Mohamed-Yasser	25-34 years old	Cegep	Yes	full stack development	VS	The amount of tools	HTC VIVE	Gaming	
I understand and agree tc	Felix Morin	18-24 years old	Cegep	Yes	3 years	Intellij	The amount of tools	Occulus Rift, Sony playsta	Gaming	
I understand and agree tc	Carlita L'Abbé	18-24 years old	Cegep	Yes	6 years	Eclipse, Intellij	The amount of tools	HTC VIVE, Google Cardb	Gaming	
I understand and agree tc	Fintan Davidson	18-24 years old	Cegep	Yes	3	Eclipse, Netbeans, Intellij,	The amount of tools	Google Cardboard, Smart	Gaming	

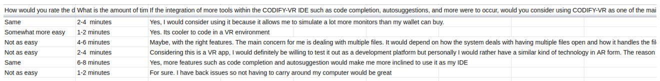
Fig. 13. Pre trial personal information survey

B. Appendix 2: Raw Data - Post-Test Questionnaire



Enter your full name (Firs	How would you rate the v	How intuitive was the app	This section concerns the	This section concerns the	This section concerns the	This section concerns the	This section concerns the	Did using the system for t	The application was easy
Felix Morin	4 Very Good	4	Neutral	Somewhat agree	Somewhat disagree	Somewhat disagree	Somewhat agree	Yes, it would be tempting	Somewhat agree
Mohamed-Yasser Housse	4 Very Good	5 Very Intuitive	Somewhat agree	Agree	Disagree	Agree	Agree	Yes. It allowed me to reali	Agree
Lara, Tran	3 Good	3	Somewhat agree	Neutral	Disagree	Somewhat agree	Neutral	Yes. It provides a fun envi	Somewhat agree
Carlita L'Abbé	4 Very Good	3	Somewhat agree	Somewhat agree	Disagree	Neutral	Somewhat agree	I like the possibility of beir	Somewhat agree
Patricia Nunes	4 Very Good	4	Somewhat agree	Agree	Disagree	Somewhat agree	Somewhat agree	Yes, it was engaging, not	Agree
Fintan Davidson	5 Excellent	3	Somewhat agree	Agree	Disagree	Somewhat agree	Neutral	It felt intriguing to do it in a	Agree

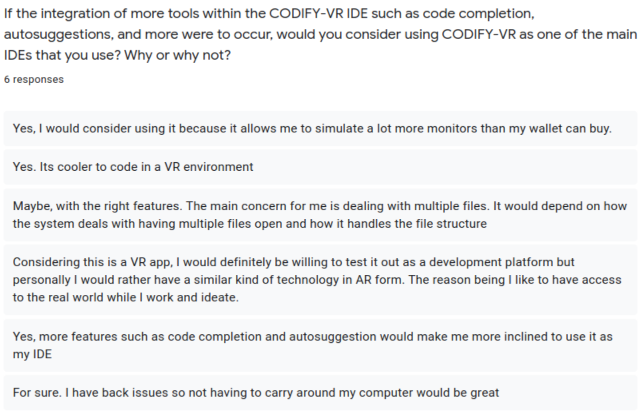
Fig. 14. Post-Trial Questionnaire raw data



How would you rate the d	What is the amount of tim	If the integration of more tools within the CODIFY-VR IDE such as code completion, autosuggestions, and more were to occur, would you consider using CODIFY-VR as one of the mai
Same	2-4 minutes	Yes, I would consider using it because it allows me to simulate a lot more monitors than my wallet can buy.
Somewhat more easy	1-2 minutes	Yes. Its cooler to code in a VR environment
Not as easy	4-6 minutes	Maybe, with the right features. The main concern for me is dealing with multiple files. It would depend on how the system deals with having multiple files open and how it handles the fil
Not as easy	2-4 minutes	Considering this is a VR app, I would definitely be willing to test it out as a development platform but personally I would rather have a similar kind of technology in AR form. The reason
Same	6-8 minutes	Yes, more features such as code completion and autosuggestion would make me more inclined to use it as my IDE
Not as easy	1-2 minutes	For sure. I have back issues so not having to carry around my computer would be great

Fig. 15. Post-Trial Questionnaire raw data Part 2

C. Appendix 3: Completed Questionnaires



If the integration of more tools within the CODIFY-VR IDE such as code completion, autosuggestions, and more were to occur, would you consider using CODIFY-VR as one of the main IDEs that you use? Why or why not?

6 responses

Yes, I would consider using it because it allows me to simulate a lot more monitors than my wallet can buy.

Yes. Its cooler to code in a VR environment

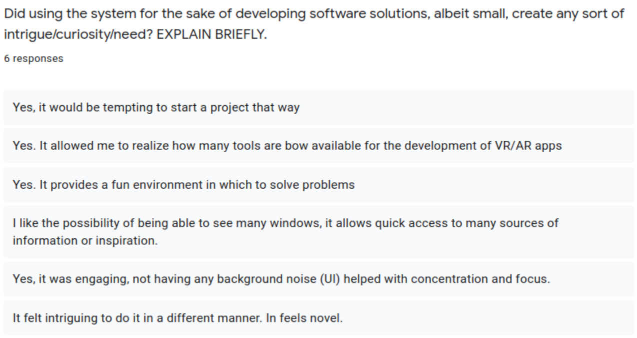
Maybe, with the right features. The main concern for me is dealing with multiple files. It would depend on how the system deals with having multiple files open and how it handles the file structure

Considering this is a VR app, I would definitely be willing to test it out as a development platform but personally I would rather have a similar kind of technology in AR form. The reason being I like to have access to the real world while I work and ideate.

Yes, more features such as code completion and autosuggestion would make me more inclined to use it as my IDE

For sure. I have back issues so not having to carry around my computer would be great

Fig. 16. Intrigue with granted the development of further tools feedback



Did using the system for the sake of developing software solutions, albeit small, create any sort of intrigue/curiosity/need? EXPLAIN BRIEFLY.

6 responses

Yes, it would be tempting to start a project that way

Yes. It allowed me to realize how many tools are bow available for the development of VR/AR apps

Yes. It provides a fun environment in which to solve problems

I like the possibility of being able to see many windows, it allows quick access to many sources of information or inspiration.

Yes, it was engaging, not having any background noise (UI) helped with concentration and focus.

It felt intriguing to do it in a different manner. In feels novel.

Fig. 17. General Intrigue

IX. References

[1] https://vr-ide.herokuapp.com/

[2] https://collections.lib.utah.edu/ark:/87278/s69w3dmf

[3] https://www.oculus.com/experiences/rift/911715622255585/

[4] https://primitive.io/

[5] https://aframe.io/docs/1.0.0/introduction/

[6] https://reactjs.org/docs/getting-started.html

[7] https://spring.io/projects/spring-boot/

[8] https://docs.oracle.com/javase/8/docs/api/