

EVALUATION OF GRASSHOPPER GOOGLE'S "LEARN TO CODE" APP

Double Vision

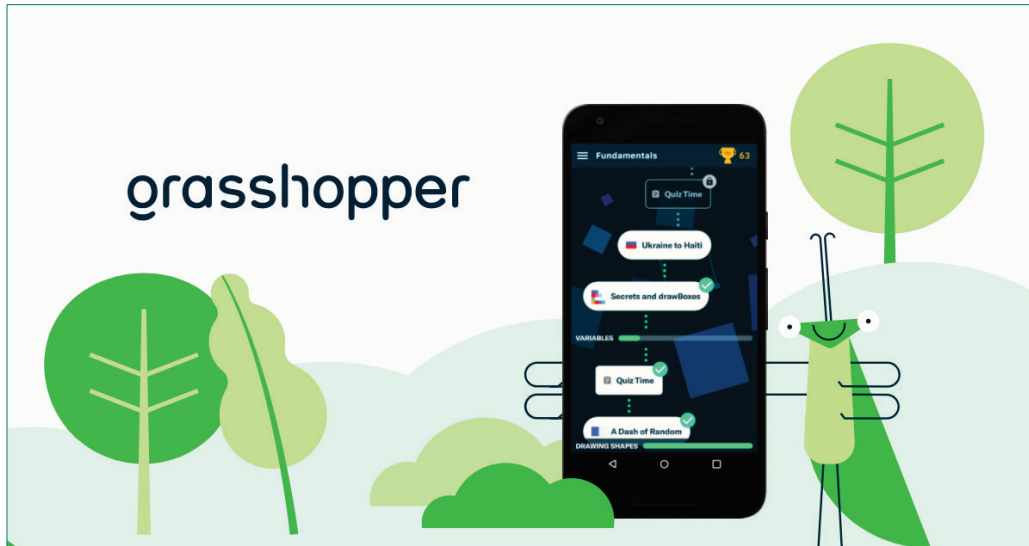
Elbert, Matt
Snider, Lingyun
Taube, Aaron
Zhao, Siyu

Table of Contents

I. PRODUCT	4
Features	5
II. AUDIENCE	8
III. INTENDED OUTCOMES	9
Breakdown of Curriculum.....	9
IV. PRODUCT CRITIQUE	10
Strength and Weakness.....	12
<i>Strengths</i>	12
<i>Weaknesses</i>	13
V. THE MATRIX.....	14
The Development of the Matrix	14
<i>Pre-Test Survey</i>	16
<i>Usability Testing</i>	16
<i>Post-Test Interview</i>	17
<i>Sentiment Analysis</i>	17
Data Reliability.....	18
VI. DATA COLLECTION PROCESS.....	18
Pre-Test Survey.....	18
Usability Testing & Post-Test Interview.....	19
Sentiment Analysis	19
VII. DATA ANALYSIS PROCESS.....	20
Pre-Test Survey.....	20
Usability Testing & Post-Test Interview.....	20
Sentiment Analysis	21
VIII. PILOT TEST	21
IX. Results and findings.....	23
Demographics Survey	23
Usability Testing	27
<i>Test Scenarios</i>	27

Interviews.....	29
Pre-Post Test Questionnaire.....	31
Sentiment Analysis	34
X. Recommendations	39
XI. Comparison and Reflections	40
XII. Contributions	41
APPENDIX	43
Participant Consent Form.....	43
Survey Questions.....	44
SECTION 1 Demographic Questions	44
SECTION 2 Pre/Post Questions	45
Usability Testing Task Scenarios	46
Interview Questions	47

I. PRODUCT



Grasshopper is a mobile/web application intended to teach the basics of Javascript and HTML/CSS to learners looking to dive into the basic foundation of programming. Grasshopper is the best way to learn to code for beginners. With fun, quick lessons on the phone or browser, Grasshopper teaches adult learners to write real JavaScript. It is currently available for free on desktop browsers, Android, and iOS. Grasshopper covers the foundations of JavaScript, how to create animations with code, teaches solving problem techniques needed for technical interviews, and introduces how to build a website with HTML and CSS.

Grasshopper is a **Code with Google** program.

Website: <https://grasshopper.app>

Developers: [Google LLC](#)

Released: April 18, 2018

Features

The latest version of Grasshopper's curriculum covers seven courses in the app: What is code, Fundamentals 1 & 2, Array methods, Animations 1 & 2 as well as Intro to interviewing, which decomposes and models real-world problems so that they can be solved by using code. The web version has two more web development courses: Using a Code Editor, and Intro to Webpages. Grasshopper gamifies the learning process of these courses in a visual way. Each lesson resembles a path with levels. Learners can collect achievements as they learn new skills. (Figure 1)

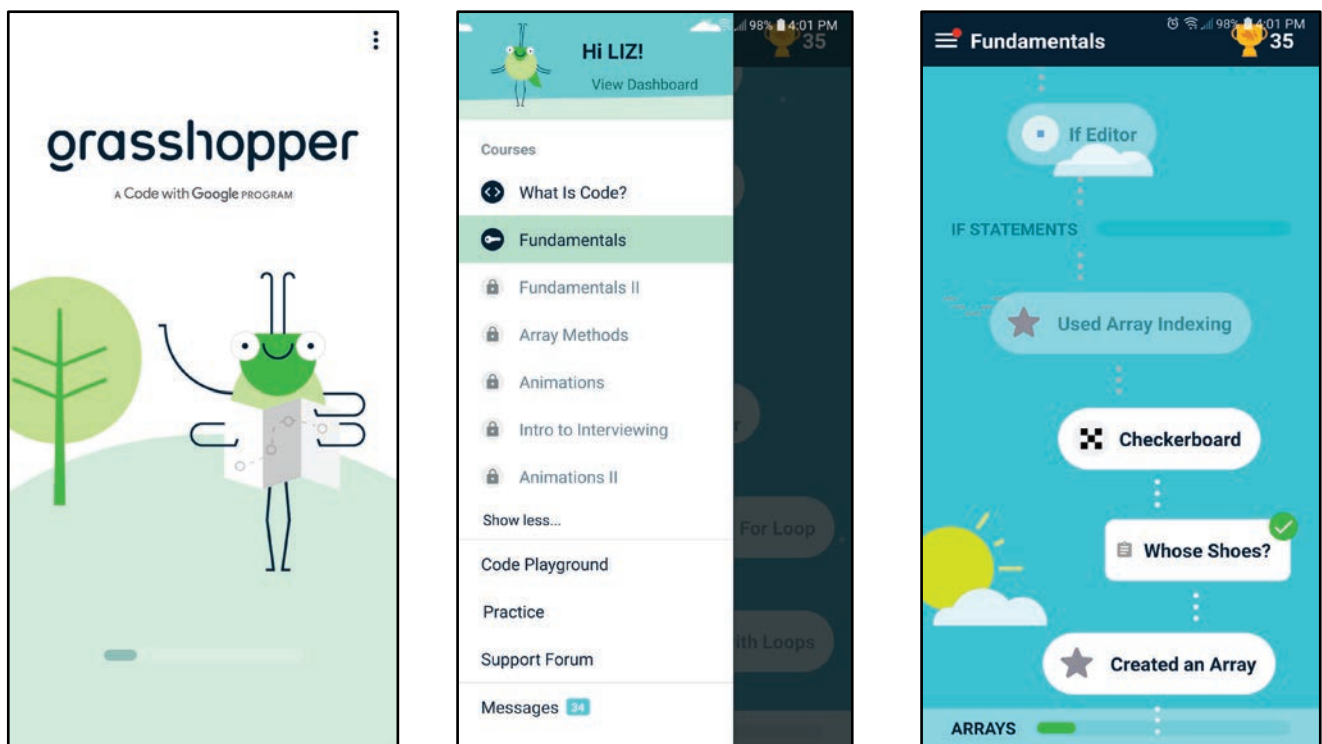


Figure 1

Each lesson is bite-sized in which the learner will be given a small puzzle to solve by following the introductions and the example solution. While providing their own solutions, the learners don't have to type out the code individually. Instead, they tap to insert the code pieces by function and

variable using the semi-visual coding method. The lessons are designed to give the learners real-time feedback so that they can see their code in action as if in a sandbox. (Figure 2)

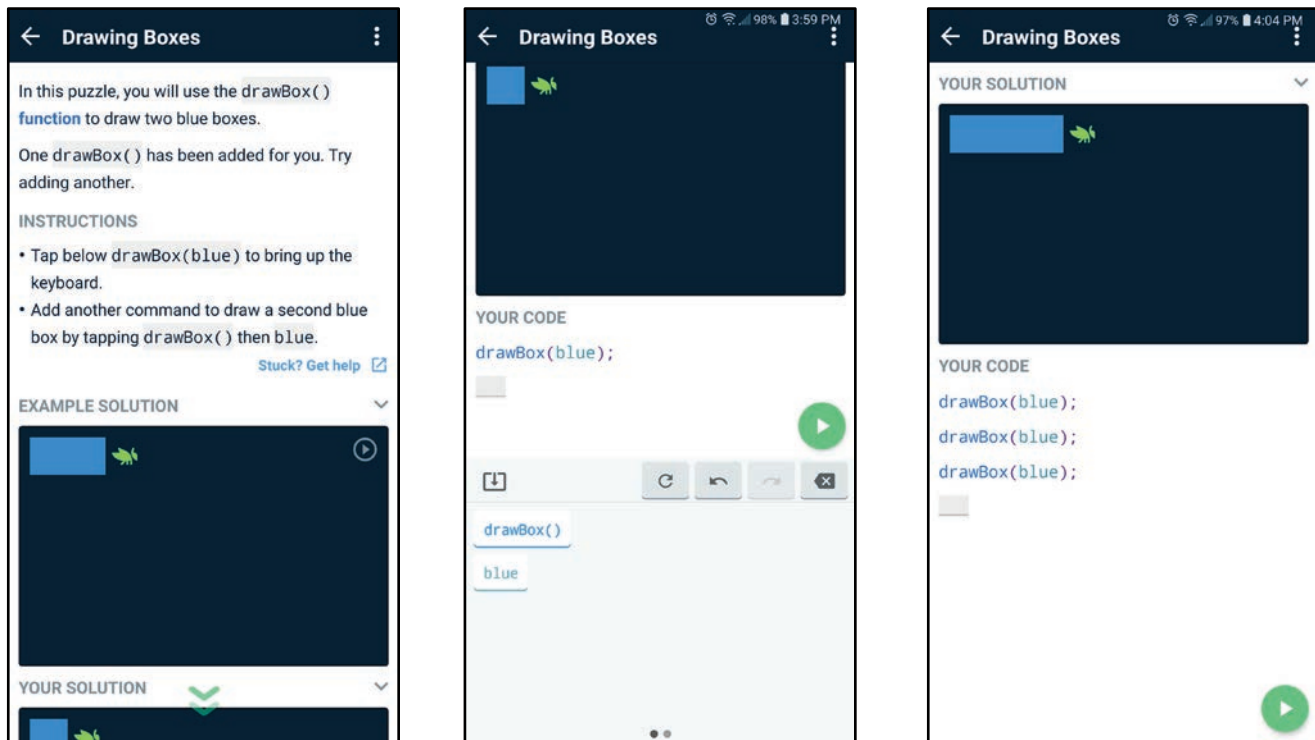


Figure 3

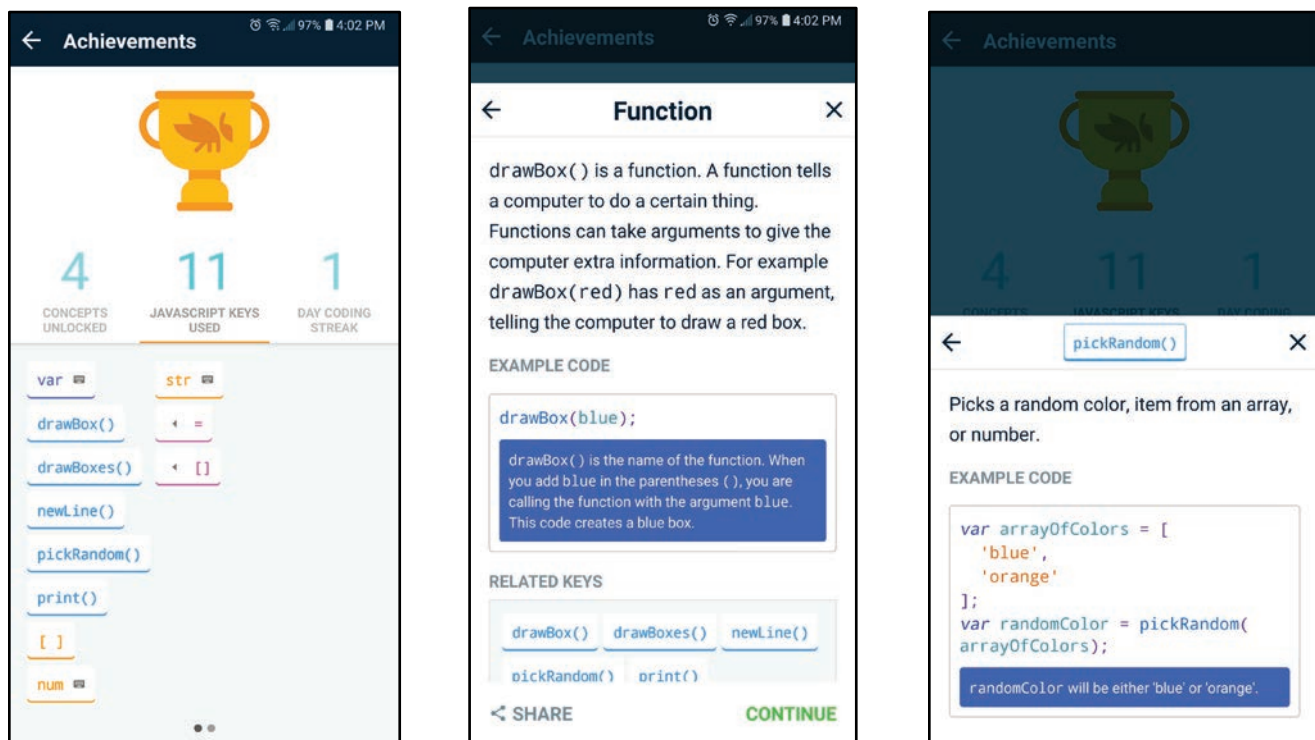


Figure 2

The achievements section is where the app summarizes what learners have learned so far: concepts such as function, string, and Javascript keys. This section allows learners to review these concepts at any time. (Figure 3)

Learners can also go to "Practice" from the menu to solve more puzzles if they need a refresher for lessons that they have learned. Or they can go to Code Playground to write their own code and see the real-time feedback on results. (Figure 4)

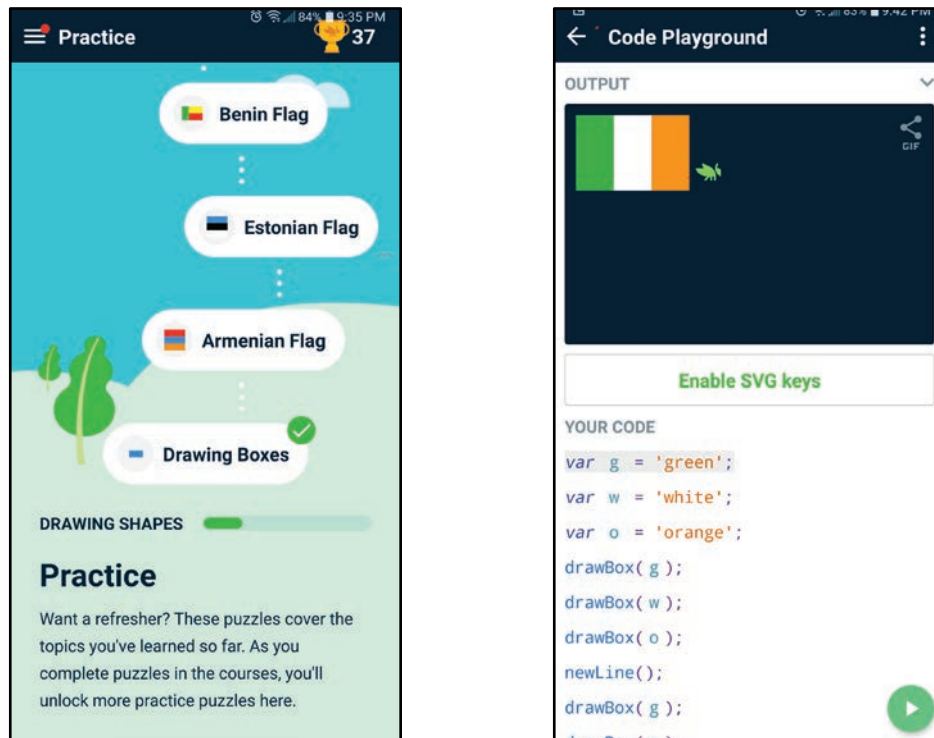


Figure 4

The app also awards Patches and Gear based on learners' achievement, sends out daily reminders to motivate and encourage them to progress in their learning journey. (Figure 5)

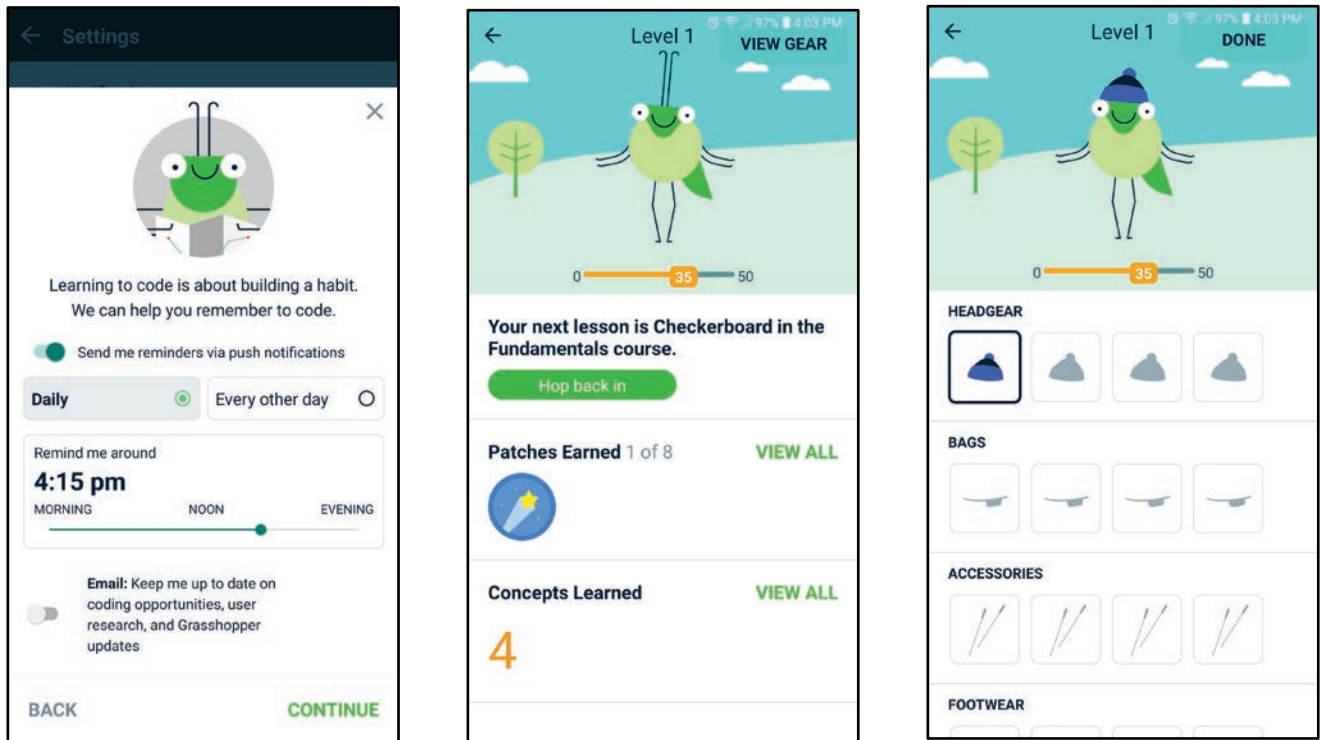


Figure 5

II. AUDIENCE

The intended audience seems to target people interested in learning coding and the age range seems to be 20-35 as "adult learners" is brought up within their content. The app store gives it a rating age of 4+ years, but that's generally more for violence guidelines and not complexity. The curriculum is based around Javascript and touches on the basics of HTML/CSS. It is marketed as "anyone can pick this up" regardless of experience. The application is geared towards people who are likely busy but still want to learn the concepts of programming - the app is marketed to individuals who are on their phone a lot and want to learn quickly and efficiently. No prior knowledge is explicitly stated for required in order to use the application successfully. Learners can expect to get a good foundation of coding and be able to see real solutions be created as they delve into the curriculum.

III. INTENDED OUTCOMES

Learners can expect to learn the Basics of Java Scripting, Algorithm Basics, Array Methods, Animations, Basics of Using a Code Editor, and Introduction to HTML/CSS Web Pages. Learners might be more effective on specific skills, possibly HTML/CSS, as that seems to be a more familiar area for this target audience than Javascript would be. The application will teach many new things for this particular learner, as it's so heavily based on acquiring new skills and knowledge. Learners should have a solid foundation of Javascript and HTML/CSS that they can continue expanding their programming skills and knowledge using this application.

The application is not marketed to be the solution that will net a job, but it will be the foundation that allows learners to go forth and further develop their programming skills. Learners will leave the curriculum program having a good understanding of why things happen on the back-end (server-side) and the solutions resulting from that on the front-end (user interface side) for mobile and desktop applications. By the end of using Grasshopper's current curriculum, they should be familiar with core programming concepts, be able to read and understand code, be comfortable solving problems with code, and be able to build a simple webpage.

More importantly, the application will increase learners' motivation and self-efficacy in coding, per Area 120, the team who developed the Grasshopper app.

Breakdown of Curriculum

Fundamentals - How code works, calling functions, variables, strings, for loops, arrays, conditionals, operators, objects, and how all these things work together.

Fundamentals II - String and array methods, creating functions, introducing recursion and callbacks, and importing libraries.

Intro to Interviewing - Decompose and model real-world problems so that they can be solved using code, utilizing computational thinking.

Animations I - Drawing shapes using the popular D3 library, defining functions, callback functions, and animations.

Animations II - Creating more complex functions using D3 and integrating JavaScript features.

Array Methods - Using multiple JavaScript array methods, such as indexOf, shift, includes, splice, map, and reduce.

Using a Code Editor - Introduction to a traditional style code editor (or IDE). Web-only.

Intro to Webpages - Covers basic HTML and CSS, then uses JavaScript to build an interactive webpage. Web-only.

IV. PRODUCT CRITIQUE

Appropriateness

The curriculum in the application is appropriate for its target audience, which is beginner coders. Each task is very straightforward. By applying visual aid, the teaching material appropriately demonstrates how coding works and how we should appropriately compose coding. Each small task links to a bite-sized piece of knowledge, and the curriculum is well structured to showcase different concepts in programming. It achieved an appropriate balance between serious coding tutorial and beginner level difficulties.

The gamification is a selling point and is informative and fun as it should be. The visual coding input method provides a comfortable entry-level experience to learners; however, it overlooks that

typing is an essential way of repeating and learning basic concepts and takes such opportunity away. This might have a potential negative impact on learner's knowledge retention.

Ease of use

The application is very easy to use. It guides the learner through a set, scaffolded path, which allows the learner to go back and review previous lessons but not skip ahead if they have prior familiarity.

Key terms are highlighted in the descriptions of the challenges and are consistent throughout. Different types of operators and variables are color-coded in what serves almost as a word bank of applicable programming concepts.

The application is satisfying to use and provides the learner with visual feedback in the form of a well-paced console printout of the learner's application running, as well as an animation of green confetti falling across the screen when the submitted program runs successfully. If the program does not run successfully, it displays a helpful hint to guide the learner. This is a good thing and a bad thing, as the learner may be running the code just to see what it does in its current state. They may not want a hint yet.

The application holds the learner's hand a lot, and they have to use specifically the tools that are given in the bank at the bottom of the screen. Concessions made to do this include requiring the learner to select from the bank when they are going to enter a string or number variable, rather than simply typing it out. This is especially more convenient for learners who use the mobile version.

There is also a "playground" that allows the learner to play in a sandbox environment and create whatever program they wish without prompt. This allows for an interested learner to experiment freely without Grasshopper directing them.

Motivational design

Grasshopper dissects the learning process into small steps and reduces the learner's frustration while increasing engagement. Details like different actions (wrong selection, hint, code piece tapping) and different states (code executing animation, object fulfilled animation) make the learner more aware of his/her engagement with the course. Such interaction provides delightful feedback for the learners to make them less likely to feel frustrated and discouraged. Their attention would be more focused on the learning process than the head-scratching debugging process in other regular programming tutorials.

Grasshopper embraces gamification in much the same style as Duolingo. It has a reward system for completing challenges, and they grow if learners maintain a daily "streak." These reward points can be spent on accessories for your Grasshopper. Push Button notifications can be enabled or disabled to remind the learner to complete a challenge if they have not already.

This system may work well with Duolingo, but it is questionable if it holds interest for learners with studying Javascript. The impact is much less apparent with learning a coding language than studying a foreign language. The learner would have to apply what they learned elsewhere to feel that they have progressed, and may find programming in another environment substantially more difficult without the provided resources.

Grasshopper also provides the learner with the option to print certificates of completion specifically to add to their online portfolios and resumes. Still, with how competitive the job market is, we sincerely doubt an employer who was familiar with Grasshopper would take this seriously for anything beyond a sophomore-level internship. Even at that point, it would be more of a way to show that the applicant takes the initiative rather than a proven skill test.

Strength and Weakness

Strengths

- For someone who has looked at code and thought, "I don't know what any of this is," Grasshopper can serve as a gentle introduction. It is reminiscent of courses that can often be tested out of at universities that introduce students to programming logic using primarily pre-done code snippets copied and pasted into a program, rather than diving straight in. The straightforward curriculum breaks down the basic concept of programming and allows learners to visualize the coding process line by line.
- Thoughtful learning mechanism including correction, hint and explanation functions that provide both gamified coding exercise and comprehensive real-world programming knowledge.
- Gamification reward elements provide an incentive for learners.
- Push Notification reminders can help a busy learner remember to take a quick daily task in order to stay the course and finish the program.
- Attractive design.
- A streamlined approach to programming eases the pain of attempting to type code on a mobile device.

Weaknesses

- Although the visual coding method helps learners who are new to coding to grasp the concept and see their code in action in real-time, it creates a disconnect between learning and application in the real world. Syntax mistakes are among the most common mistakes that programmers make. Without the opportunity to make such mistakes during learning, learners would miss an important component of the coding process - debugging. It is also essential for programmers to get familiar with real code editors, the coding habits such as commenting and consistent indentation, so their code is readable and manageable. These

practices cannot be taught in Grasshopper mobile version. The code editor is only available in the web version and is locked until learners complete several courses.

V. THE MATRIX

The Development of the Matrix

As part of **Code with Google** program, the Grasshopper app aims to make it possible for people to get into coding in a fun and easy way. We wanted to validate if the intended outcome has been achieved.

Our team met over Zoom to discuss what we would measure and how. Ideally, for a learning-focused app, we wanted to measure the first three levels per the Kirkpatrick model: Reaction, Learning, and Behavior. However, with the limited time and test participants we had access to, it was not realistic to evaluate if the learners have retained and applied the knowledge they learned by using the app. So we focused on the first level and measured their learning experience. We brainstormed the following possible questions to evaluate:

- Is the app -
 - Usable? (Usability)
 - Useful? (Effective for what it intends to do)
 - Used? (Overall app experience that encourages user adoption)
- As a learning app
 - Is the content relevant?
 - Will the learners have a hard time following?
 - Do the learners enjoy learning via the app?

- Is it easier or harder for them to learn via this app versus going to an instructor-led course or watching YouTube?
- Does learning from the app increase the learners' confidence to continue learning to code?

We analyzed the list and grouped them into five categories: usability, learning material, motivation mechanism, usefulness, and overall user satisfaction. Then we rewrote them into six questions that we will be using in the matrix. (Table 1)

QUESTIONS	INSTRUMENTS			
	Pre-test survey*	Usability Testing	Post-test Interview**	Sentiment analysis
	Demographic survey + Pre-test questionnaire ¹		Interview questions + Post-test questionnaire ²	
1. Is the app easy-to-use for the learners? (Usability)		X	X	X
2. Is the material in the app easy-to-follow for the learners? (Learning material)		X	X	X
3. Does the app motivate the learners to learn, and how? (Motivation mechanism)		X	X	X
4. What experiences does the app offer that the traditional programming courses can't? (Usefulness)			X	
5. Is the app useful for learners to learn coding? (Usefulness)		X	X	X
6. Would the learner recommend this app to others? (User satisfaction**)			X	X

Table 1

* Pre-test survey does not answer these questions directly but will provide background information of participants.

** User satisfaction is measured by NPS (Net Promoter Score), which asks only one question of how users will recommend the product to their friends and family.

^{1,2} *Pre-test questionnaire in Pre-test Survey and Post-test questionnaire in Post-test Interview use the same set of questions for a pre/post comparison.*

To evaluate these questions, we chose the following instruments: **pre-test survey, usability testing, post-test interview, and sentiment analysis.**

Pre-Test Survey

The pre-test survey can provide both quantitative and qualitative data and included **two parts**: a demographic survey and a set of pre-test questions.

The demographic survey allowed us to better understand the background characteristics of our tryout participants. It can give us insights on whether the differences in characteristics would impact participants' experiences and performance during the testing and how those factors could have potential influence.

We reached out to the development team that is behind Grasshopper, Area 120, during the research. Heather Smith, the Technical Program Manager of Area 120 shared that they also aim to increase students' motivation and self-efficacy (their own perception of their abilities) in coding through using the app, while teaching them some foundation coding skills. So we decided to include a set of pre- and post-test questions in our tryouts to evaluate if the app is useful in increasing self-efficacy. The pre-test questions are the second part of the pre-test survey, while the post-test items are the same questions asked during post-test interviews. We would collect the "before & after" data from the pre- and post-questions for further analysis.

Usability Testing

We planned to utilize contextual inquiry as the usability study method in which the researcher watches the user interacting with the object they are studying and discusses these interactions with the user during the process. Compared to other observation methods, the contextual inquiry would require

our usability test participants to perform the tasks and explain what they are doing and what they are thinking at the same time. This would allow us to see the entire usage situation and context of use, understand participants' exact behavior, and gain insights on which other learners might be involved when and why. After conducting a pilot test, in which we noticed that the participant did not feel comfortable about "showing and telling" simultaneously, we changed contextual inquiry method to observation for the official tryouts so that the participants were not required to explain their activities while performing tasks. This adjustment of the usability testing method should not impact the results.

Post-Test Interview

After the usability testing, we would conduct a post-test interview with the participants and ask them a series of summative questions about their overall experiences. Considering the high-resolution data we could gather from the contextual inquiry, post-test interviews would help us encapsulate those data and patch our understanding of the participants' behavior during the contextual inquiry.

As mentioned earlier, during the post-test interview, the same questions we used in the pre-test survey (part II) would be asked again for us to collect the post-test data on the participants' perception of coding and learning coding.

Sentiment Analysis

It is essential to collect data elsewhere to make sure that we could decrease the likelihood of data inaccuracy and reliability due to the limited sample size that we would be conducting testing on, which could potentially skew the results and decrease the power of this research. Sentiment analysis can give us an overview of the broader public opinion and see the sentiment behind each area we aim to evaluate. The data we collected for sentiment analysis is the existing app users' reviews from both

the Google Play store and the iOS app store. We would examine the patterns detected from the data and use them to triangulate the findings with other data sources.

Data Reliability

The method upon coming up with the questions for the contextual inquiry and post-test interview was based on the overarching goal of gaining insights on questions we listed in the matrix.

When developing these questions, we used the following guidelines to ensure that the questions were written in a way that is easy for participants to give accurate and complete answers.

- Make questions as short as possible without confusion.
- Eliminate redundant questions.
- Use the vocabulary that is appropriate for the target users - beginning level coding learners.

Avoid using phrases that are too technical.

- Avoid writing too general or ambiguous questions, leading questions or biased questions.
- Avoid asking two questions in one.
- Propose questions according to the participant's interactions.

VI. DATA COLLECTION PROCESS

Pre-Test Survey

The demographic survey was created and posted on SurveyMonkey for its usability and capabilities. Data collecting would also be conducted via this platform. The survey took about 5 minutes for participants to complete.

The pre-test questionnaire was created in Google Forms along with the post-test questionnaire (same questions, but two forms). We had to utilize Google Form as a free SurveyMonkey account only allows for 2 surveys.

Everyone in the Double Vision team ensured that their participants completed the demographic survey and pre-test questionnaire before the usability testing.

The questions for both demographic survey and pre-test questions are included in the appendices.

Usability Testing & Post-Test Interview

Usability testing, followed by a post-test interview, was also conducted with every participant. They were done in different set-ups based on the participant's choice and what devices they had access to. The virtual sessions were conducted via Zoom, and the participants were asked to share his/her screen for us to observe their actions. Also, we recorded the sessions for further analysis. We also had one participant who preferred an in-person session in which we followed the safety protocols to eliminate any possibilities for disease transmission. The in-person session and the mobile screen were video-recorded as well. The entire process took about 30 minutes for each participant, from set-up to completing the interview.

Each team member of Double Vision conducted the usability testing and post-test interview with one participant. The questions for usability testing and interviews are included in the appendices.

Sentiment Analysis

Sentiment analysis of Grasshopper did not involve our tryout participants. It scrapes the app reviews and ratings posted by existing app users in both the Google Play store and the iOS app store. Lingyun Snide was responsible for collecting data for this analysis by utilizing a service called Appbot.

This tool enabled us to collect real-time data directly from both app stores by simply searching and locating the Grasshopper app.

VII. DATA ANALYSIS PROCESS

Pre-Test Survey

We used descriptive statistics to provide patterns summarized from the demographic data of the pre-test survey on demographic, attitudinal, and lifestyle indicators of the participants. This was used to determine if the participants were a representative sample of the target population. Given the small sample size of targeted users, we used SurveyMonkey's built-in charts to visualize the results.

The data from the second part of the pre-test survey - a 6-question pre-test questionnaire - was analyzed along with the data collected in the post-test questionnaire in Google Forms.

Usability Testing & Post-Test Interview

We observed the participants' effort to complete the prescribed tasks and rated the perceived difficulty of the tasks on the following scale.

- 1- Task completed Smoothly
- 2 - Minimal difficulty
- 3 - Moderate difficulty
- 4 - Significant difficulty
- 5 - Task not completed

We interviewed each participant about their experiences and reaction to the app. Areas where participants were observed experiencing difficulty had their researcher's notes examined to find possible reasons for the problem. It was challenging to determine how frequently this issue would

occur in reality due to the limited number of usability tests we were able to conduct. However, we could cross-examine this with our sentiment analysis to identify sources of negative or even positive sentiment and therefore project just how widespread this problem was.

We utilized the Likert Scale to analyze the answers to the same six pre/post questions collected before and after the test to determine efficacy and engagement. Using a Likert scale, we could measure a change in a participant's perceived ability to use or learn coding from using Grasshopper. The participant should rate their categories such as confidence level in learning coding before the test (within the survey) and then again afterward (during the interview). The difference in user score before and after would indicate the application's effectiveness in providing benefits along these key categories.

Sentiment Analysis

Before discovering Appbot, we planned to analyze sentiment data by passing them through a NLP (Natural Language Processing) in Google Cloud Platform and extracting entities (words and phrases), classifying those entities into "topics" such as satisfaction, complexity, design, etc, then analyzing them in Excel or Tableau. Appbot allowed us to skip these steps and directly receive the sentiment scores, word/phrase lists, topics and emotion chart. We were also able to customize the tool to get insights on aspects that we could not evaluate due to time and resource constraints, such as effectiveness.

VIII. PILOT TEST

Our tryout test plan for each participant is as the following (Table 2):

When	Who	What
10/27-11/2	Researchers	Each to find at least one participant for the learner's tryout. (Each researcher will conduct testing with the participants they found individually.) Criteria: the participant should have little to none experience in coding.
11/2-11/11	Official tryouts	
	Participants	Sign the consent forms
	Participants	Fill out the demographic survey - 5 min
	Participants	Fill out the pre-test questionnaire - 5 min
	Researchers	Set up usability testing equipment (AV, Zoon, recording, etc.) - 5-15 min
	Researchers	Explain to the participants about the purpose and methods of the testing - 5 min
	Researchers and participants	Help participants to get the app installed and logged in, if needed - 0 - 10min
	Researchers	Ask participants to perform the three tasks from "Usability Testing Task Scenarios," observe their actions and rate on the Observation and Interview worksheet (Appendices) - 10-20min
	Researchers	Interview participants (interview questions in Appendices) and make notes of their answers on the worksheet - 5-10 min
	Participants	Fill out pre-test questionnaire - 5 min

Table 2

We conducted a pilot test with one volunteer participant using the list above and was able to complete it within 60 minutes. We have not seen issues such as blank or inconsistent answers to questions, or responses to open-ended questions without understanding the questions. However, the participant was not quite used to the contextual inquiry and did not feel comfortable about "showing and telling" simultaneously. Therefore, we changed contextual inquiry to observation for the official tryouts so that the participants were not required to explain their activities while performing tasks.

IX. Results and findings

Demographics Survey

We used SurveyMonkey to create a demographic survey for all participants to take before the usability test. We were able to gather data from the survey directly on SurveyMonkey and the *Analyze Results* feature within the tool made it simple to create data visualizations for these specific questions.

Overall, our demographic survey findings were in-line with the expected and proclaimed target audience based on our market research on the Grasshopper App. The majority of our participants were not aware of the app when the survey was conducted, and over half had little to none experience in programming, a prime target audience for users of the app. Our participants are individuals who are for the most part every day regular people familiar with technology, and thus no "tech professionals" were a part of our study. They are between the age of 23-30, most have a college education and happened to be all male.

The following is a detailed analysis of the demographic survey.

Question 1: "When learning, what approaches are most helpful to you?" (Figure 6)

This was a multi-select question and (100%) answered that *Hands-On (Learning by doing it myself)*, (75%) selected *Demonstration (someone showing me)*, (50%) selected *Watch videos*, (25%) selected *Reading (instructions, manuals, books)* and (0%) chose *Other*. Gathering that 100% preferred to do hands-on learning was useful information to collect from this question. We know that the Grasshopper App's main action of learning is by doing this, so that was a positive result from this question.

Question 2: "Have you heard of the Grasshopper App before?" (Figure 7)

This was a Yes/No question. All of our users (100%) selected that they *had not heard of the Grasshopper App before*. This was a positive result as we can determine from this that we will have unbiased results based on their usability test and their pre/post-test surveys.

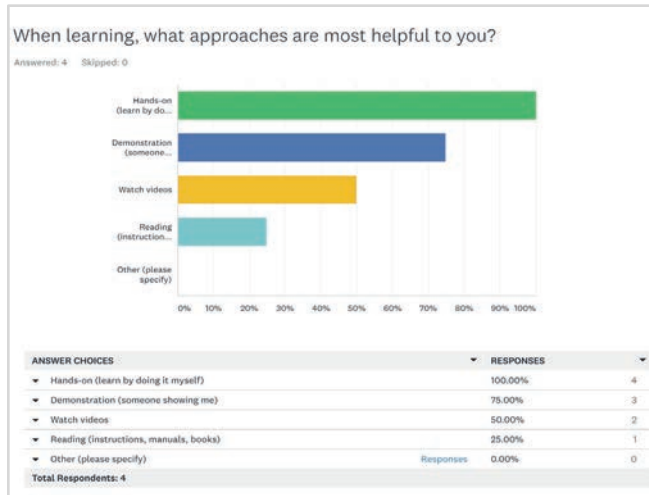


Figure 7

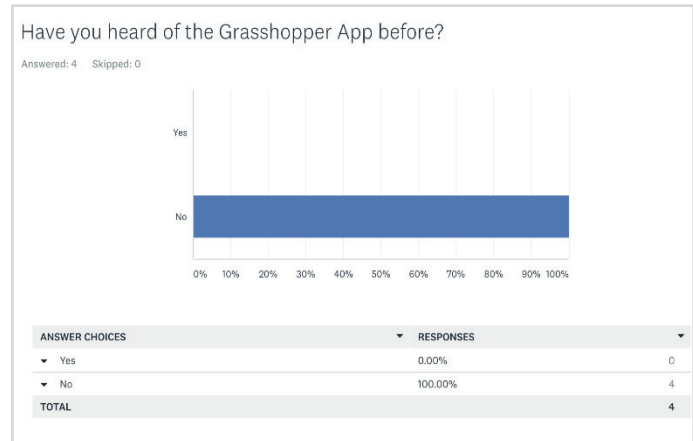


Figure 6

Question 3: "How experienced with programming would you say you are?" (Figure 8)

This was a multiple-choice question. Our highest score was (50%) of which selected *Below Average*, (25%) selected *Average*, and (25%) chosen *No Experience*. Overall, this is not surprising and aligns well with who our target audience is: young adults and no experience - below-average experience is what we would expect and hope to gather for this test and analysis as that is the group of which the app is meant for.

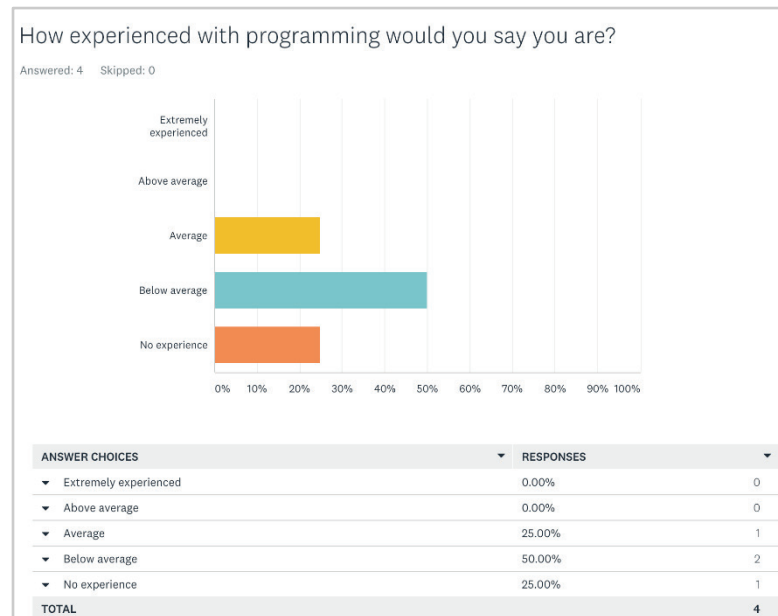


Figure 8

Question 4: "What is your age range?"

(Figure 9)

This was a multiple-choice question. (100%) selected that they were *between the ages of 23-30*, and the rest were all (0%). Our analysis for this data is that it's spot on for the type of user that the Grasshopper App targets with its marketing and its actual educational product and curriculum.

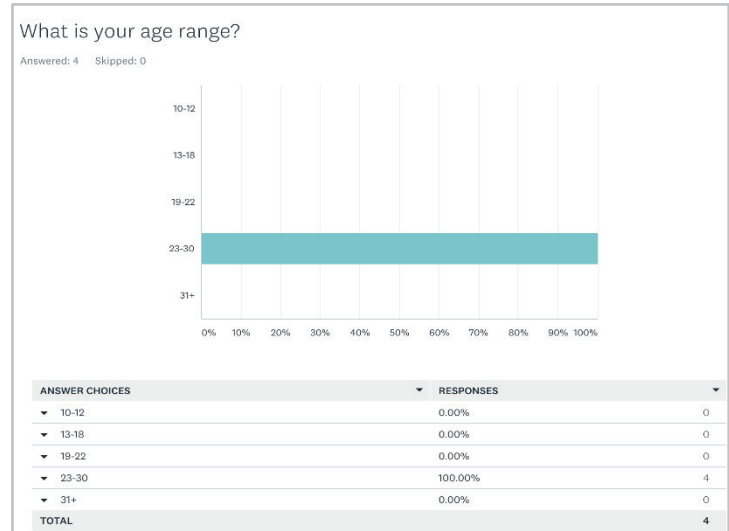


Figure 9

Question 5: "What is your gender identification?" (Figure 10)

This was a multiple-choice question. (100%) selected *He/Him* and (0%) for the rest. We were a bit surprised that there was not a little more diversity, but with the small pool of participants, it was understandable as the majority of software engineering or programming roles tend to fall within the male gender. We do not believe that this would create bias or impact our results, however.

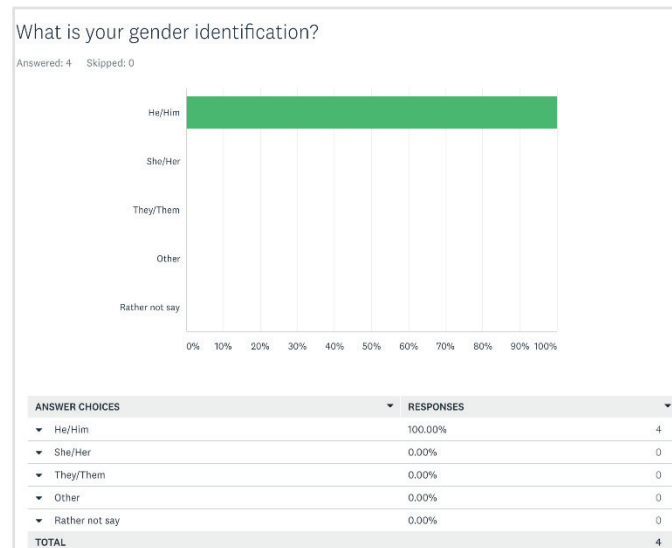


Figure 10

Question 6: "What is your education level?" (Figure 11)

This was a multiple-choice question. (50%) selected *Undergraduate Degrees*, (25%) chose *High School Degree*, and (25%) selected *Master's Degree*. We believe this data is not surprising as the Grasshopper App is truly meant to target individuals who do not necessarily have any specific level of education. The majority of young adults (the target audience) tend to have undergraduate degrees, so for that to be the majority of the users made sense and aligns with our expectations.

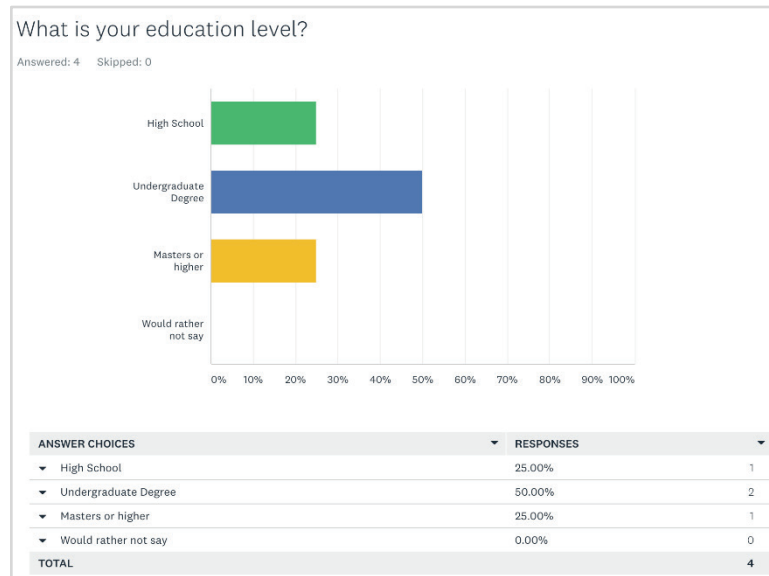


Figure 11

Question 7: "How would you rate your tech-savviness?" (Figure 12)

This was a multiple-choice question. (75%) selected *"I'm aware of new tech, but I don't seek out the latest and greatest, but I can use it if I have to"*, (25%) selected *"Up to date on the latest and best apps and technology - I can quickly figure out new tech by myself."* and (0%) selected *"I don't know about new technology until someone else tells me about it - I usually need some help."*

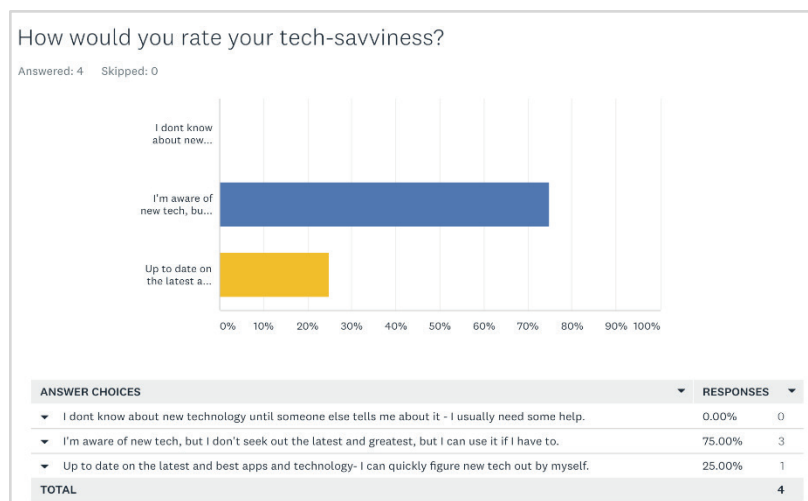


Figure 12

selected *"I don't know about new technology until someone else tells me about it - I usually need some help."*

help." This data point overall aligns with the expectations for our target audience as most of the participants were (23-30) and thus having (75%) makes the most sense for the answer that was selected for this question.

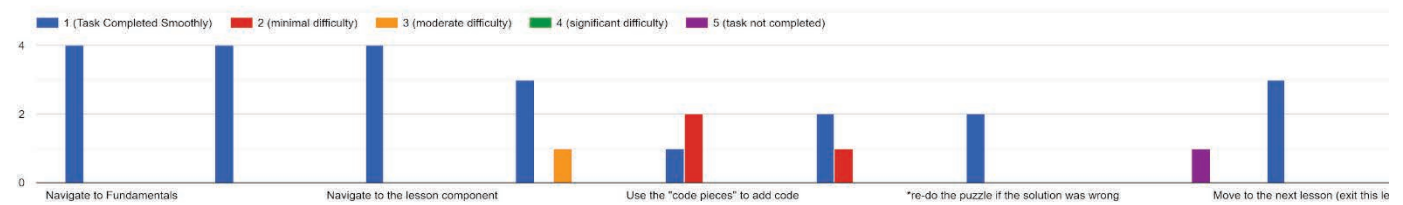
Usability Testing

Our participants were subjected to a basic usability test of Grasshopper. In this test, they navigated from the initial login to the first two fundamental lessons, as well as through some additional features. The purpose of this test was to measure whether participants with different levels of experience with Grasshopper could complete the associated tasks for our usability test. Per the data collected, our participants found Grasshopper to be mostly intuitive, except for needing some improved guidance for the core mechanics of the application. Also, the participants thought Grasshopper would be useful for programming learning, especially for people with limited experience. Due to its simple interactions, immediate feedback, and hands-on tutorials, Grasshopper was regarded as an easy-to-follow instructional application.

Test Scenarios

For the first fundamental test, the login and navigation to the first lesson were intuitive, and all participants got to it without fail. One of our 4 participants experienced difficulty understanding the on-screen instructions for the lesson and did not continue. Two of our three continuing participants stumbled trying to use the code pieces to complete the first lesson, and one did not find the "run" button right away. In the second lesson, the two still willing participants did not experience any trouble at all repeating these behaviors to solve the new challenge. This would indicate that while the application may not provide perfect guidance on how to begin coding in lessons, once the user understands it, the code piece controls are simple to follow. (Figure 13)

Find the first lesson under Fundamentals and take that lesson.



Take the next lesson if the participant is willing to.

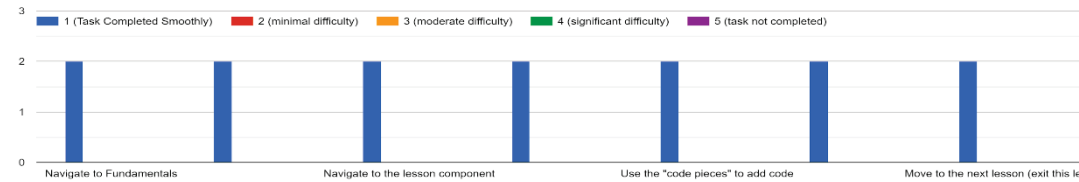


Figure 13

Progress tracking features are mostly intuitive, but some difficulty realizing the trophy icon is interactive, and the place to look for achievements was had with this test. (Figure 14)

Set a daily reminder to take lessons in this app

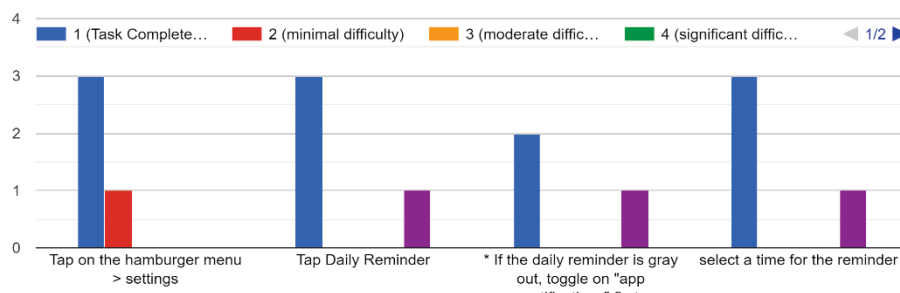


Figure 14

Daily reminders to take app lessons were overall easy to set up once the user was made aware of its existence. Only one user had any signs of difficulty accessing settings, meanwhile, a separate user ran into an interesting problem of the option not existing, it was not even visibly greyed out. It is not yet known why the option did not show, but this would be a cause for concern for a user. (Figure 15)

Show me what you have achieved

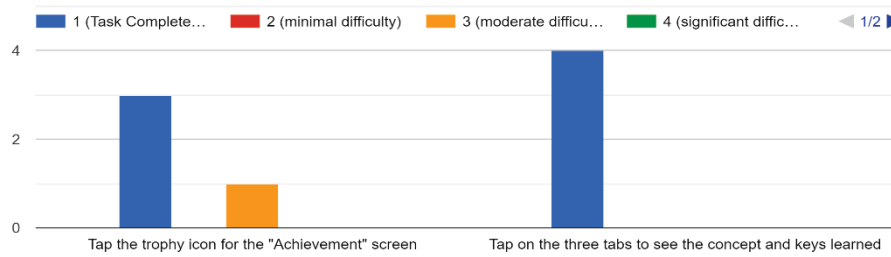


Figure 15

Interviews

In general, all participants believed that Grasshopper would help people who plan to learn coding but without much experience. Among the 4 participants, 3 of them thought Grasshopper would be very useful, and 1 of them thought Grasshopper would be useful. (Figure 16)

3 of 4 participants thought Grasshopper was simple and straightforward to start with, while 1 of them said they had some difficulty at the very beginning. This finding aligns with our recommendation that Grasshopper might want to create better guidance on how to start the lessons. (Figure 17)

As you currently understand Grasshopper, how would you rank its potential usefulness as a learning tool for entry-level programming?
4 responses

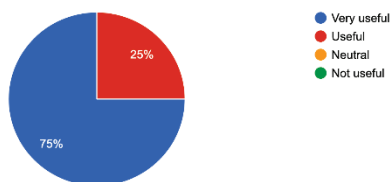


Figure 17

Did you feel the lesson you took was easy to follow?
4 responses

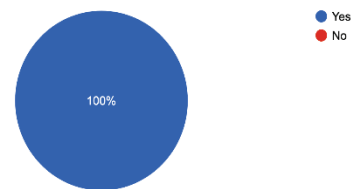


Figure 16

All participants claimed that the lessons on Grasshopper were very easy to follow. The intuitive tutorial, immediate feedback, and selecting buttons rather than typing codes to program were regarded as the main reasons behind it. However, one participant also said there was no basic understanding of what each component did, making the content much harder to understand.

All participants found the tool engaging and stated that they wanted to keep going and see what else they could do. One stated that it was similar to a fun puzzle-solving experience for them. The motivation system in Grasshopper encourages the participants to cover more lessons.

For the most part, users did not experience significant difficulties using the application. One had some trouble finding the "Achievements" because they did not associate the trophy icon with this type of feature. Another thought that the first lesson was a bit confusing in that it restricted the sequence in which they could complete the lesson, and they desired to do it differently. The third participant who got stuck noticed that there was a hint button to tap but felt that they wanted to figure it out without the help. In that case, we assume Grasshopper might want to improve its initial tutorial to walk the users through the basic features to avoid frustration.

Our participants mostly did not have prior experience with code-learning tools, with one exception. That participant had previously used "Scratch," a block-based programming tool, which they thought did not do as good of a job as Grasshopper in starting the user off slow and making sure they knew where to start.

For improving Grasshopper, all of our users felt that navigation and functionality did not require improvement. Two thought it could use some UI and UX improvements, both in terms of look and feel, as well as general usability. One thought it needed to have more extensive coursework options. (Figure 18)

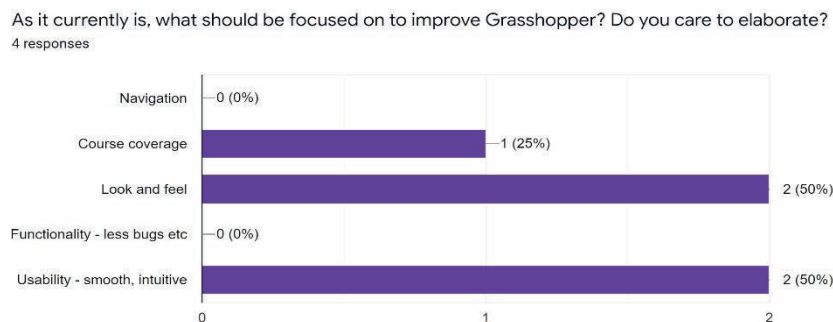


Figure 18

All our users would recommend Grasshopper to others, and all agree it is best for children or people with absolutely no programming experience. Two participants even thought that elementary school age kids could enjoy the experience as well as middle to high school aged kids. (Figure 19)

After trying the app, would you recommend Grasshopper to others?
4 responses

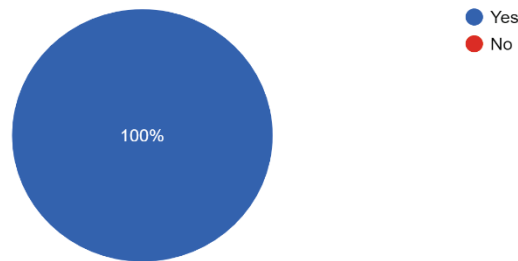


Figure 19

Pre-Post Test Questionnaire

A pre-test and post-test analysis were conducted via Google Forms. They consisted of Likert Scale questions, and both tests contained the same six questions. Each participant was given the survey, but due to technical difficulties, one set of responses was lost. Of those remaining, we found that participants on average had increased confidence in their ability to code, but a decreased interest in learning to code.

The first question, "I have an interest in learning to code." turned out a reduced interest in the post-test. This does not necessarily mean anything wrong about the app, as it may have been a low commitment way for someone to discover they just are not that interested in programming. (Figure 20)

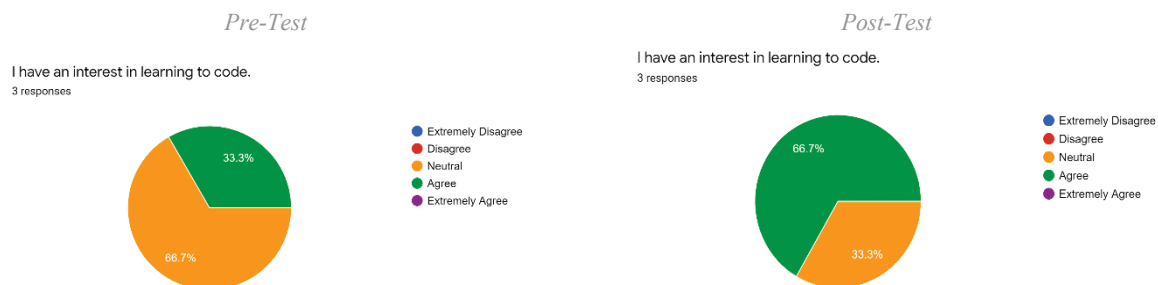


Figure 20

Participants did not find coding any more or less difficult after using Grasshopper. This isn't entirely revealing of anything as the users only completed one lesson - we can speculate that there would be more "Disagree" results if users worked through the full course-work. (Figure 21)



Figure 21

Despite not believing coding was any easier after using Grasshopper, all participants did believe they could learn how to code afterward, indicating that Grasshopper did not make coding seem easy, but did make it seem more approachable. (Figure 22)



Figure 22

Our participants did not see themselves as working as a programmer, though one was neutral to the notion. After using Grasshopper, they all had moved to neutral, instead of disagreeing with the notion that they could see themselves working as a programmer. (Figure 23)

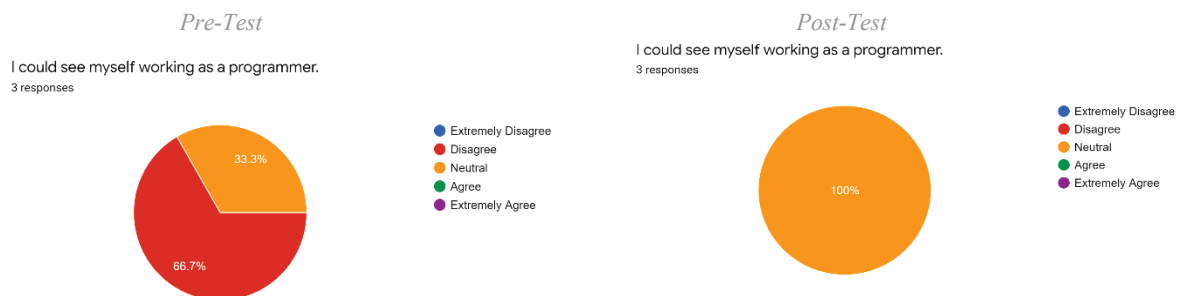


Figure 23

Participants actually lost interest in the Grasshopper mobile application post-test. One participant who chose "Agree" in the pre-test chose "Neutral" in the post-test. This is consistent with our user that lost interest in learning how to code. We believe there would be more variance in these responses if the participants thoroughly worked through the entire course curriculum. (Figure 24)



Figure 24

We had an increase in our participants' belief that coding can be fun, despite one having lost interest in coding. This indicates positive signs of how engaging the design of Grasshopper is. (Figure 25)

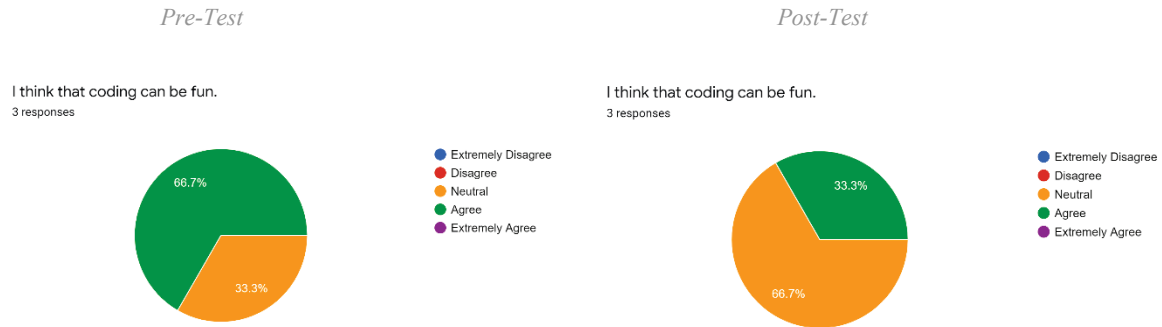


Figure 25

Sentiment Analysis

At the time of data collection for sentiment analysis, 15,999 reviews and 59,725 ratings were scrapped from the Grasshopper app reviews in both Google Play Store and iOS app store and passed into Appbot for future study. The goal was to detect patterns from the data and gain insights on usability, complexity, design, usefulness, and user satisfaction. We found that the app users are satisfied with Grasshopper and consider it effective, easy-to-use, and fun for beginners to learn programming.

The overall sentiment score is 90% on a scale of 0-100%, and the average ratings in both stores are 4.5 and more. (Figure 26)

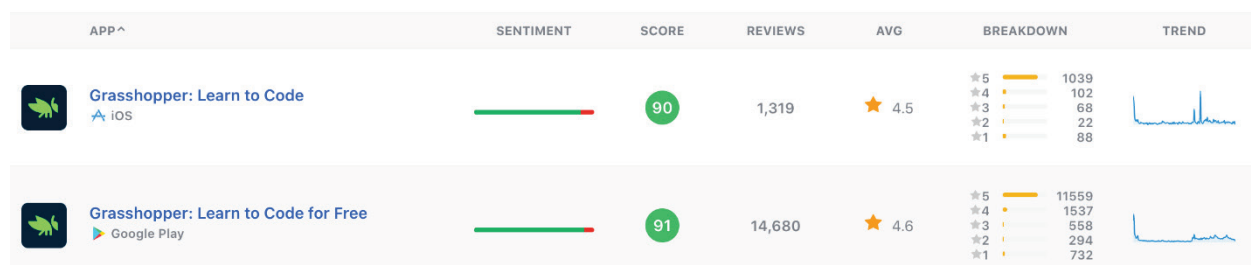


Figure 26

The emotion chart below also indicates high overall user satisfaction. (Figure 27)

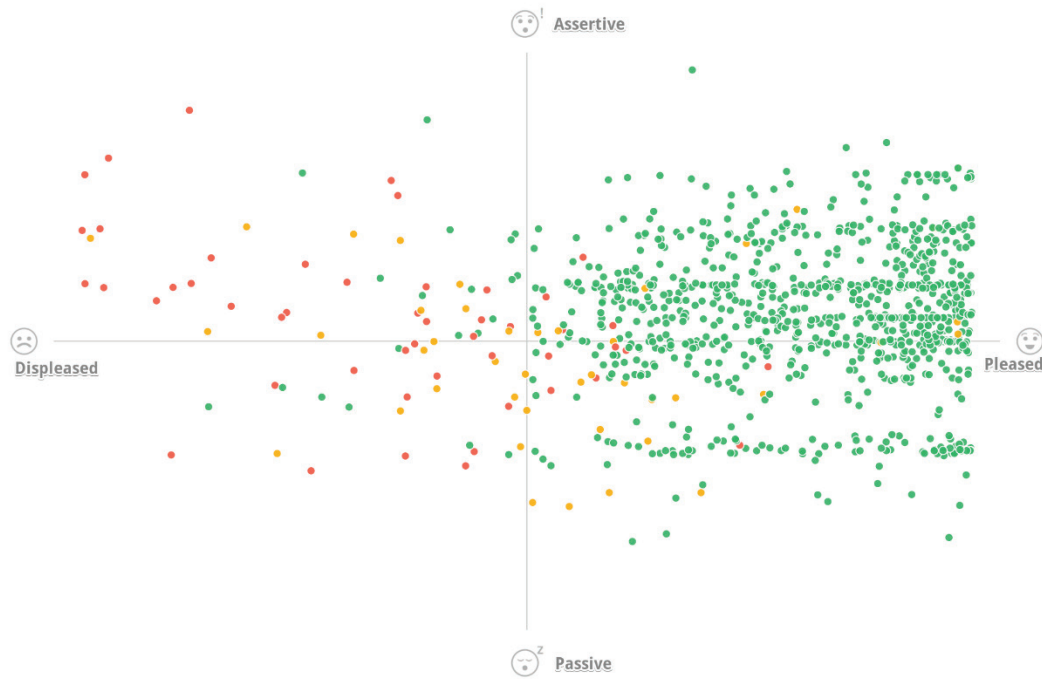


Figure 27

The entities (words and phrases) of each review were extracted and classified into topics, which provided us a breakdown of what users' reviews and ratings were for. From the following top topics, we can see that the user experiences are among the most reviewed aspects. (Figure 28)

	TOPIC	SENTIMENT	MENTIONS	OVERALL REVIEWS	TREND
1	Satisfied users	<div><div></div></div>	8,314	63.1%	
2	Complexity	<div><div></div></div>	2,127	16.1%	
3	Design & UX	<div><div></div></div>	1,974	15.0%	
4	Use cases	<div><div></div></div>	1,774	13.5%	
5	Internationalization	<div><div></div></div>	541	4.1%	
6	Bugs	<div><div></div></div>	516	3.9%	
7	Performance	<div><div></div></div>	459	3.5%	
8	Feature Requests	<div><div></div></div>	428	3.2%	
9	Customer Support	<div><div></div></div>	381	2.9%	
10	Dissatisfied users	<div><div></div></div>	187	1.4%	

Figure 28

The following images show the top words and phrases extracted from the reviews (by popularity - number of times they appeared) as well as the sentiments associated with them. With these lists along with the word cloud, we could detect a pattern which indicates that users consider Grasshopper to be fun, easy to use and helpful when it comes to teaching beginners how to code, and the experiences are pleasant. (Figure 29-30)

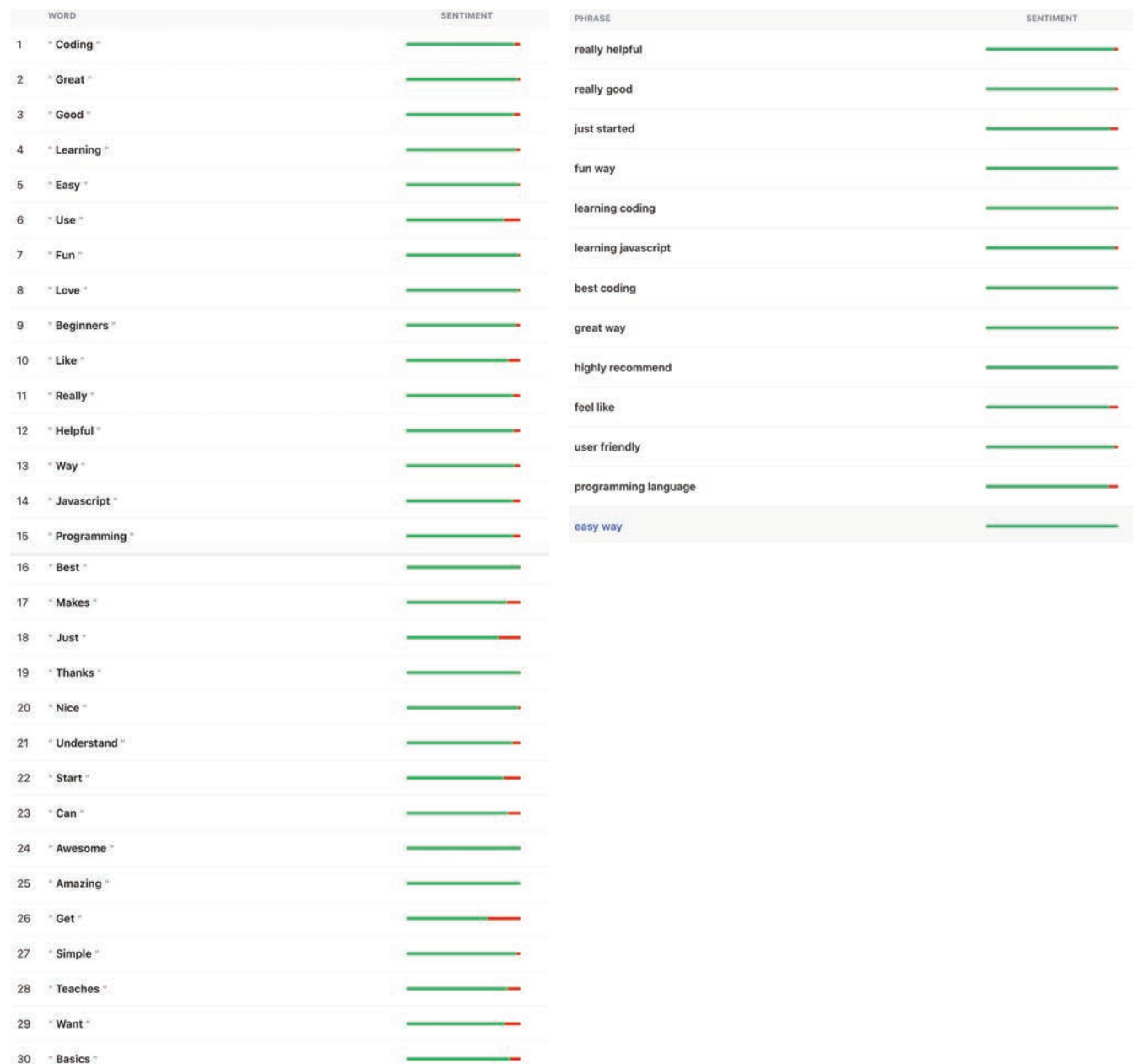


Figure 29

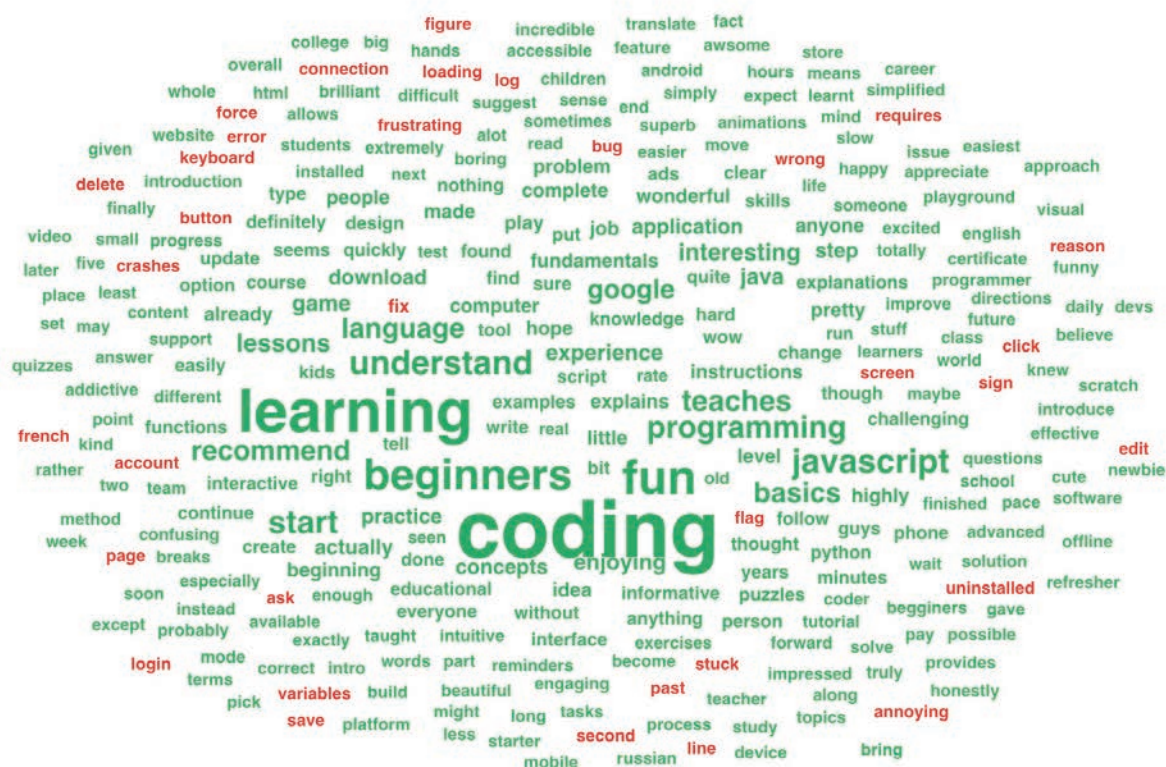


Figure 30

Although Google states that the intended audiences for Grasshopper are adult learners, the app did receive the [Best EdTech award](#) in 2018 and was recommended for kids aged 6-12. We did not have access to children while conducting the user testing but did find relevant data in the reviews. There were 292 reviews that specifically mentioned that the app was great for teaching kids to code. (Figure 31)

One of the aspects we would like to measure but did not include in our tryout plan due to the time and resource constraint was learning effectiveness. Out of curiosity, we looked into the review data and discovered that 945 reviews mentioned the effectiveness of the app, in which 92% of them were positive comments.

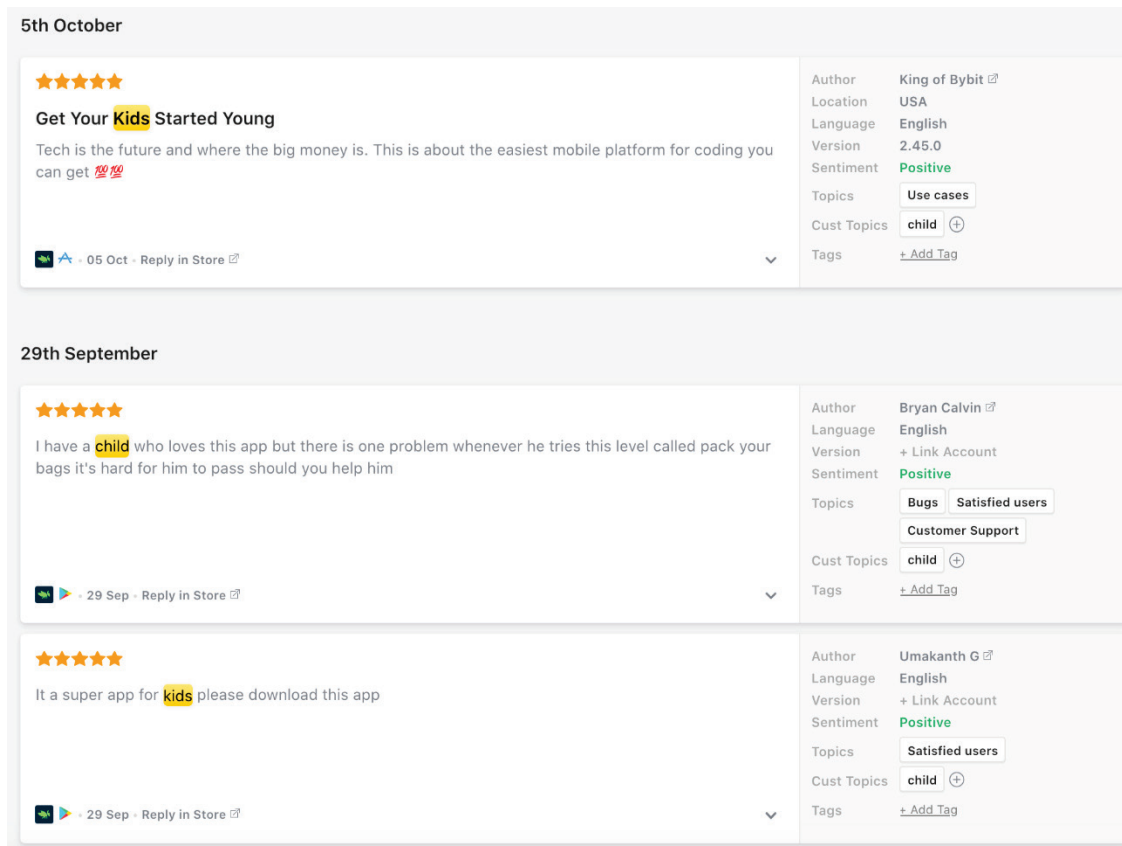


Figure 32

During our initial evaluation, we pointed out that one of the weaknesses of learning via Grasshopper that we saw was the gap between learning and real-world application. Although only a few, we did see the same concern from users. For example, in the following review, the user thought the lesson was wrong as he/she could not replicate the results by typing the coding learned in Grasshopper into a standard editor. (Figure 32) This was due to the fact that all code in Grasshopper only works in their own IDE (integrated development environment). But the reasons behind are too complicated for a beginning coder to comprehend; therefore users would think the lesson itself was at fault.

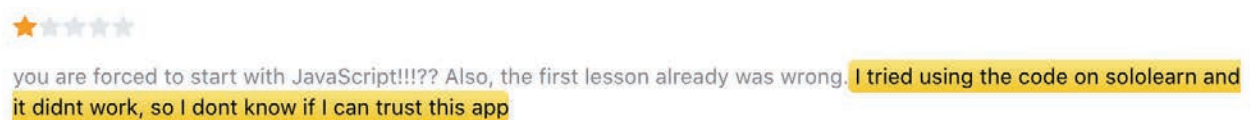


Figure 31

X. Recommendations

Given the objective of Grasshopper, which is to motivate learners and boost their confidence in coding while teaching some foundation skills, we conclude that the app has accomplished it based on the findings from our tryouts and the sentiment analysis of app users' reviews. Our recommendations are as the following:

Add an **in-app guided tour** and more visual cues so the absolute beginners can start their first lesson in Grasshopper with ease. In our tryouts, a couple of participants had some difficulty with the instructions on how to get started. But once they did, they could proceed with other lessons without issues. An in-app guided tour or walkthrough with an option to skip will resolve this type of difficulty.

Add the **landscape orientation** option. Our own tryout participants did not have such experiences. However, according to some reviewers, Grasshopper locks the phone screen rotation in portrait orientation, which breaks long code into multiple lines, including arrays and character combinations. This produces orphan brackets, parentheses, and strings, making it challenging to read and understand certain code, especially for beginning coders. A landscape orientation could eliminate most of these frustrations.

Expand **course coverage**. Both our participants and the app users mentioned that they would like to see more extensive coverage of programming lessons beyond just Javascript, HTML, and CSS. Although on Grasshopper's website, the team states that they have not yet had a plan to expand the curriculum within the app, the [Code with Google](#) program* does offer multiple learning paths if learners were to advance their programming skills.

** Grasshopper is part of the Code with Google program under the "Self-directed learning" section.*

XI. Comparison and Reflections

The learners' tryouts' results mostly aligned with our initial evaluation in terms of the app's usability, competitiveness, motivational design, and usefulness. We also gained insights on a couple of aspects that we did not cover in the initial evaluation, such as the appropriateness of other audiences than the adult learners, effectiveness, and user satisfaction.













	Initial Evaluation	Learner Tryouts & Analysis
Usability	 <ul style="list-style-type: none"> • Easy to use • Attractive design 	 <ul style="list-style-type: none"> • Straightforward • Would like to have a better guidance
Complexity (learning material)	 <ul style="list-style-type: none"> • Easy to follow - hand-holding • Bite-sized learning 	 <ul style="list-style-type: none"> • Easy to follow • Intuitive tutorial • Would like to see more course options
Motivation mechanism	 <ul style="list-style-type: none"> • Gamification • Motivational design: Patches & gear 	 <ul style="list-style-type: none"> • Engaging - users want to keep going <ul style="list-style-type: none"> ◦ progress pathway; fun puzzles
Usefulness	 <ul style="list-style-type: none"> • Unlock the door to programming • Push notification 	 <ul style="list-style-type: none"> • Yes as a learning tool for entry-level, high school students, kids • Boost confidence and interest in coding • 292 reviews said "great for children"
User satisfaction		 <ul style="list-style-type: none"> • Will recommend to others • Sentiment score 90% (scale of 0-100%) • 4.5 average rating (scale of 1-5)
Effectiveness	 Gap between learning and real-world application	 92% positive in 945 reviews that mentioned Effectiveness

Table 3

Perhaps because of our backgrounds as programmers and designers, we did not initially believe that the onboarding process of Grasshopper was insufficient, but considering the difficulties that the participants had, clearly more can be done to help the learner.

While it has its intended purpose, which is primarily teaching Javascript, with the number of in-demand languages out there that are also easy to pick up, it is within reason to desire additional course options such as Python.

The motivational design of the app engaged users to keep going after they successfully completed one lesson which was a fun puzzle game. Although not being able to test the long-term impact, the reward system – achievements, patches, and gears – also encourages users to learn in the long run, according to app users' reviews.

Our study found the usefulness of Grasshopper to be as a point of entry to coding primarily, and it can have the effect of increasing user's efficacy and interest in code development. Also, it was observed that participants were able to figure out their further interest in coding with limited commitment via trying Grasshopper.

Also, both the participants and the app users stated that Grasshopper was easy to follow and encouraging. They also believed that it would assist people who would like to learn programming but without too much experience. Generally, they were willing to recommend this tool to others, which indicates that the users are very satisfied with Grasshopper.

Lastly, we did not include effectiveness in either our initial evaluation or the tryouts. However, the sentiment analysis showed that app users do consider Grasshopper an effective app for beginners to learn coding.

XII. Contributions

During the evaluation project, each team member of Double Vision shared responsibility and contributed to the following tasks:

- Brainstorming and ideation
- Report writing and editing
- Allocating test participants and conducting user testing
- Data collecting

- Presentation

Also, each team member has led the following work:

- Aaron Taube: pre/post questionnaires for the tryout plan; data analysis for usability testing, interviews, and pre/post questionnaires.
- Lingyun Snider: usability testing scenarios; usability test worksheet and questionnaire set-up (Google Forms); sentiment analysis; team coordination and formatting/ submission of the report for each phase.
- Matt Elbert: demographic survey questions, set-up (SurveyMonkey) and data analysis.
- Siyu Zhao: data analysis for usability testing and interviews.

APPENDIX

Participant Consent Form

I agree to participate in the study conducted by **Iowa State University - HCI Program**.

I understand that participation in this usability study is voluntary, and I agree to immediately raise any concerns or areas of discomfort during the session with the study administrator.

Please sign below to indicate that you have read and you understand the information on this form and that any questions you might have about the session have been answered.

Date: _____

Please print your name: _____

Please sign your name: _____

Thank you!

We appreciate your participation.

Survey Questions

SECTION 1 Demographic Questions

1. When learning, what approaches are most helpful to you?
 - ☐ Hands-on (learn by doing it myself)
 - ☐ Demonstration (someone showing me)
 - ☐ Watch videos
 - ☐ Reading (instructions, manuals, books)
 - ☐ Other (please specify) _____
2. Have you heard of the Grasshopper App before?
3. How experienced with programming would you say you are?
 - ☐ Not at all experienced
 - ☐ A little bit of experience
 - ☐ Moderately experienced
 - ☐ Expert experience
4. What is your age range?
 - ☐ 10-12
 - ☐ 13-18
 - ☐ 19-22
 - ☐ 23-30
 - ☐ 30+
5. What is your gender identification?
 - ☐ He/His
 - ☐ She/Hers
 - ☐ They/Them
 - ☐ Other
 - ☐ Would rather not say
6. What is your education level?
 - ☐ High school
 - ☐ College
 - ☐ Master or higher
 - ☐ Would rather not say
7. How would you rate your tech-savviness?

- Early adopter- Gets the latest and greatest apps and technology- can quickly figure new tech out by oneself.
- Middle of the road- aware of new tech, but doesn't seek out the latest and greatest- can use it if you have to.
- Novice- doesn't know about new technology until someone else tells them about it- might need help to use new technology at first.

SECTION 2 Pre/Post Questions

Read each question carefully and rate on a scale of 1-5 how much you agree or disagree. 1 being strongly disagree, 5 being strongly agree, and 3 being that you have no opinion or are neutral.

1. I have an interest in learning to code.
2. I believe learning to code is difficult.
3. I believe that I can learn how to code.
4. I could see myself working as a programmer.
5. I have an interest in the Grasshopper mobile application.
6. I think that coding can be fun.

Usability Testing Task Scenarios

Task:	1 (Task Completed Smoothly)	2 (minimal difficulty)	3 (moderate difficulty)	4 (significant difficulty)	5 (task not completed)
Find the first lesson under Fundamentals and take that lesson.					
Navigate to Fundamentals					
Find the first lesson (French Flag)					
Navigate to the lesson component					
Understand the on-screen instructions					
Use the "code pieces" to add code					
Tap the "Run" button to execute the code and see results					
*re-do the puzzle if the solution was wrong					
*use "reveal solution" to see the correct solution code					
Move to the next lesson (exit this lesson)					
Take the next lesson if the participant is willing to.					
Navigate to the lesson component					
Understand the on-screen instructions					
Use the "code pieces" to add code					
Tap the "Run" button to execute the code and see results					
*re-do the puzzle if the solution was wrong					
*use "reveal solution" to see the correct solution code					
Move to the next lesson (exit this lesson)					
Show me what you have achieved					
Tap the trophy icon for the "Achievement" screen					
Tap on the three tabs to see the concept and keys learned					
Set a daily reminder to take lessons in this app					
Tap on the hamburger menu > settings					
Tap Daily Reminder					
* If the daily reminder is gray out, toggle on "app notifications" first					
select a time for the reminder					

Interview Questions

1. As you currently understand Grasshopper, how would you rank its potential usefulness as a learning tool for entry-level programming?
 - Very useful
 - Useful
 - Neutral
 - Not useful
2. How difficult or simple was it to know where to begin?
3. Did you feel the lesson you took was easy to follow?
4. What aspects of the app made learning more simple or more difficult for you to learn the content?
5. Did you feel motivated to proceed to the next lesson after you completed one lesson? Why or why not?
6. (If noticed that the participant had difficulty at a certain point) I see you "xxxxxx" (running into difficulties), could you please explain more about how you see the problems and your attempt to resolve them?
(Otherwise)
At any point did you get "stuck"?
 - If you did - did you use any of the features to get support?
 - If not - why is that?
7. Have you tried other apps for learning coding?
 - If "Yes", what does Grasshopper offer that other apps don't?
8. As it currently is, what should be focused on to improve Grasshopper? Do you care to elaborate?
 - Navigation
 - Course coverage
 - Look and feel
 - Functionality - less bugs etc
 - Usability - smooth, intuitive
9. Who do you think Grasshopper is best for, from what you've seen?
10. After trying the app, would you recommend Grasshopper to others?