A Concise Reference App For Skateboarders

Brady Hagen
bhagen@oxy.edu
Occidental College

1 Introduction

Skateboarding has skyrocketed in popularity since its initial hayday all the way back in the 1960s [7]. As Famous Skateboarder Tony Hawk said, "I consider skateboarding an art form, a lifestyle and a sport." [6]. From its humble beginnings of wooden crates with metal bearings bolted on, to becoming its very own professional sport and making its debut at the Tokyo Olympic Games. While seventy years of history, regional lingo, and convoluted tricks is exactly what gives skating its counterculture-esque allure, it also makes it intimidating for newcomers. I propose lowering the barrier to entry for newcomers by creating a comprehensive glossary of various skateboarding terms, tricks, styles, and techniques. This glossary will be packaged as an app for viewing on the go and serve as a handy reference book for beginners and veterans alike. The information in the app will also follow a unique organizational structure, with heavy inter-linking between terms that lead to pages nested within one another.

1.1 Problem Context

Skateboarding is unique in that there isn't winning or losing. While there are competitions and ways to challenge your friends, skating mainly involves working on personal-development. Winning in a sense could instead be defined as getting better and progressing. When a person gets into skateboarding they taste the sweet nectar of victory a lot. At the very beginning just figuring out how to balance on a board, then learning how to push properly, and then riding all on your own feel great. After that, then what? Some people view skateboarding as just a means to get around, it's cheaper than a bike or car and its small size makes it convenient to carry around. For others, riding is just the beginning, the real meat and potatoes is found in the tricks.

What you can do with a piece of wood with wheels is extensive. Without handlebars, a seat, or anything else to get in your way you're granted an extraordinary degree of freedom. For some this is liberating, for others it's paralyzing.

Choice paralysis isn't a new phenomenon. It came to prominence in the early 2000's. Columbia psychologists Sheena Iyengar and Mark Lepper presented dozens of different kinds of jams to shoppers at a local supermarket. There they found unique behaviors in shoppers when they were presented with twenty-four flavors of jam, and then only six flavors of jam. When presented with more choices it was found that people not only make worse decisions, lament about the options they didn't choose, but often end up making no choice at all [2].

The psychological phenomenon of choice paralysis extends outside of preserved fruit to skateboarding as well, as the amount of options and versatility are immense. Each trick has a unique term, which changes depending on the trick before it, the way you're facing when doing it, which part of the board you're on, and which of your feet did which part of the trick. Knowing what things are even called is another hurdle for newcomers. As mentioned above, the naming scheme in skateboarding is far more complex than one would expect. The degree of control offered to you while riding a skateboard means that there really does have to be a name for every little thing in order to communicate what it is you're trying to do. A Backside 360 to Frontside Noseblunt Side might sound like a load of gibberish to you, but to the judge's scoring Street League Skateboarding it's the highest scoring trick of all time. Different tricks also mean entirely different things depending on where you live. For example a 'Casper Heel', 'Hospital Heel', 'Scissor Flip', and 'Flower Flip' all refer to the exact same trick in which you catch the board upside down midway through a heelflip and backside shuvit-ing out[5].

The effects of choice paralysis are also compounded by the frustration and difficulty inherent to skateboarding. Skateboarding is difficult, as part of the main draw is finally landing that thing you've been working on for weeks and getting a rush of dopamine and adrenaline that makes it all worth it. But it can be hard to pick something that then becomes what you decide to do for the next twenty hours of your life. On top of this the way skateboarding tricks are often communicated to beginners is that tricks build off of one another. In order to learn how to Kickflip, you need to learn how to Ollie. Every trick builds off of the skills learned before it. This compounded nature means what you'll be able to do in the future is decided entirely by the now.

Lastly, almost all the information available on skateboarding is spread out across several different platforms, mediums, and with debatable authenticity. Many such sources take the form of massive never-ending lists which give you every single trick alphabetically sorted in a giant unmanageable blob, others look like they were written in 2004 by a troubled teen in detention. Looking up what an 'Ollie' is will bombard you with information and fifteen minute long YouTube videos, yet the more niche the trick the less reliable information there is, and you have to resort to Reddit comment sections. Information is spread out, un-reliable, hard to filter and sort through, and lastly not concise. The last thing you'd want to do while enjoying a skate session with your friends is sit down for several minutes watching a YouTube video.

The solution I propose involves utilizing video platforms like YouTube, as a means to solve this information gap using related videos, while also offering times within videos where the relevant content is found. A user would insert an image of an object in real life or even a drawing (from their lecture notes for example), and my project would return videos with related images either in the thumbnail or the video frames. If related video frames are found within the video, times corresponding to these frames will additionally be returned.

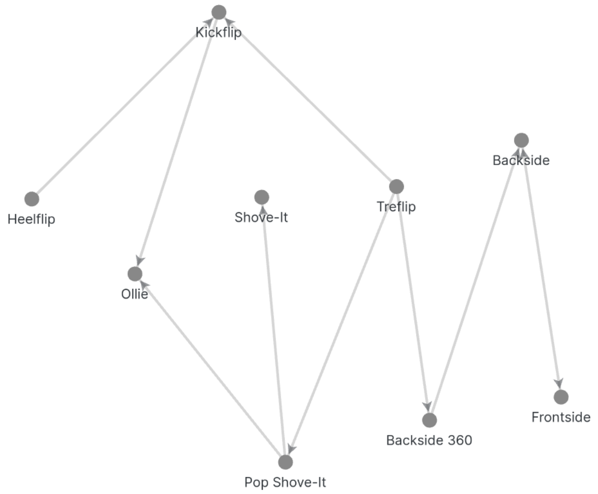
YouTube videos can be broken down into images through both the thumbnail, an image preview of the video content, and the individual frames which make up the same video. With my project, we essentially create "baskets" of videos which relate to each other by visual characteristics (called "features"). Upon inserting an image, finding relevant information means finding the basket of images (that refer to videos or video frames) which relate to our input image by comparing features. By utilizing video frames as opposed to simply other images (like thumbnails), my project speeds up the searching process by returning a time within the video. From an educational perspective, not only does this help students fulfill these information gaps through videos, but it can speed up the studying process as users don't always have to sift through a video to grab the relevant information.

2 Technical Background

2.1 Organization

The core of the app will be the unique way it displays information. Each skateboarding term added will include a definition, a video of the trick being performed in real life, and a simple diagram showcasing feet placement. The goal of this is to provide everything you could need or want to know about a trick at a quick glance so you can get back to the fun part of skateboarding. This on its own is serviceable, as having all information centralized and in one place with visual aids is helpful enough. However, due to the interlinked nature of skateboarding tricks it requires prerequisite knowledge that most beginners won't have. For example, you might not know what a 'Heelflip' or a 'Backside' is, and thus wouldn't understand when I tell you that a 'Tre Flip' is a combination of a 'Backside-360 Pop Shove-it and Heelflip'. For all you know I just made up some words and smushed them together. You would then have to open up a browser and look up what each of those words are in order to understand your initial request. This would defeat the purpose of the whole app in the first place if every term couldn't be understood by beginners.

The answer to this problem comes in the form of creating densely linked definitions. Let's go back to the term 'Treflip'. The definition I provided was that it's: 'a combination of a Backside-360 Pop Shove-it and Heelflip'. So in order to understand what I'm talking about, you'd need to know what a Backside is, a Backside-360, a Shove-it, a Pop Shove-it, a Kickflip, and a Heelflip. All the prerequisite knowledge quickly adds up. You could replace each term with their own definitions but imagine how bloated and humongous each definition would be, and on top of that it would work fine until you try to communicate with someone else who skateboards. They'd be using terms like Backside and Pop Shove-it, not whatever literal translation the app would've been using. Instead, I propose using a form of nested linking. Each term that could be potentially confusing for newcomers would be underlined, and when interacted with would expand and provide their definition. Any and all terms in that definition would also be underlined and could be interacted with to list out their definitions as well. Anything confusing links to its definition and anything confusing in that definition links to its definition. This removes any hierarchical structure and allows organization to occur through meaning and association.



Pictured above is a graph view for all the Terms that make up the term 'Treflip'. It illuminates how each word builds and relates to one another. While the graph view is helpful in how it illustrates relation, it certainly isn't the best way to showcase information. Instead of presenting terms as nodes on a graph, we'll opt for a more linear style of presentation. This more linear style would only showcase one term at a time, and when a link to a new term is clicked it would simply appear below the first term. This would create a simple way to navigate and understand relation without creating any crazy confusing structures.

The benefits inherent to this type of system of linking are massive and will be fully utilized by the app. Niklas Luhmann, a prominent German sociologist famous for his work on Systems theory and Social theory coined the term 'Zettelkasten' in 1992. A Zettelkasten was a slip box full of index cards that Luhmann used to take notes. He would later credit the box as an independent partner who helped him write his 70 books. What made his box of cards unique was its strong emphasis on linkage between ideas and concepts. Luhmann writes: "In any case communication becomes more fruitful when we succeed to activate the internal network of links at the occasion of writing notes or making queries. Memory does not function as the sum of point by point accesses, but rather utilizes internal relationships and becomes fruitful only at this level of the reduction of its own complexity. In this way, more information becomes available at this isolated moment of a search impulse than one had in mind. There is also more information than was ever stored in the form of notes. The slip box provides combinatorial possibilities which were never planned, never preconceived, or conceived in this way." [4] The basic idea here is that as we research and think about terms or tricks we begin to see a relation form between various concepts and terms.

Vannevar Bush, in his piece "As We May Think", writes "Our ineptitude in getting at the record is largely caused by the artificiality of systems of indexing. When data of any sort are placed in storage, they are filed alphabetically or numerically, and information is found (when it is) by tracing it down from subclass to subclass. It can be in only one place, unless duplicates are used; one has to have rules as to which path will locate it, and the rules are cumbersome. Having found one item, moreover, one has to emerge from the system and re-enter on a new path. The human mind does not work that way. It operates by association. With one item in its grasp, it snaps instantly to the next that is suggested by the association of thoughts, in accordance with some intricate web of trails carried by the cells of the brain." [1]

If we push ourselves to add lots of links between our notes, that makes us think expansively about what other concepts might be related. This allows us to not only retain but learn at a much more efficient rate.

2.2 App Development

There are lots of ways to make a mobile app. However, due to the unique design goals for our glossary app we will have to be selective in which technologies we decide to use.

The app should be lightweight so one can pull it up quickly for reference and close it just as fast. This means no logins, no ads, and bloat should be kept to a minimum.

Our app should be cross-platform. If the goal of the app in question is to make information about skateboarding uniform and more accessible to everyone, ensuring that it's available on both iOS and Android platforms is essential. Instead of doubling development time by learning both native development for Android Studio and XCode for Android and iPhone's respectively, we'll take the code one time, deploy everywhere approach and look for frameworks that enable us to deploy our code base everywhere. The design and layout is extremely important, as the philosophy of the app is to present information in a concise and easy to understand manner. Because of that, it's important we make our own layout and not rely on a pre-built template.

With these specifications in place it leaves us with two main frameworks for creating apps, Flutter and React Native. There are plenty of similarities between these two frameworks. Both utilize cross-platform development, both have tons of support and development from massive companies, they have an extensive add-on and extension community, and both are very welcoming for newcomers with their in-depth documentation and on-boarding process. Although React Native isn't truly native as it wraps its elements in a shell many consider it to be native regardless. Because of this both boast native capabilities in terms of performance and design. The main difference between Flutter and React Native is the design architecture and language. Flutter uses Dart, React Native uses JavaScript. Due to the time limitations in place on this project and the fact that I'm already familiar with both JavaScript and with React, I've thus chosen React Native over Flutter.

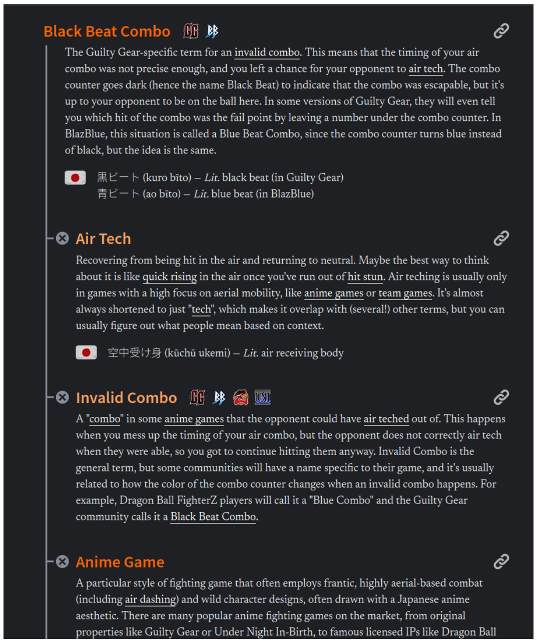
There are nearly as many options for the back-end as there are for the front-end. The two most popular back-end services to pair with React Native are Google's serverless Firebase, and NodeJS paired with a database of choice like MongoDB. The main difference between the two is that Node requires one to actually handle server and back-end functionality. Which would be helpful if our project was ultra reliant on superb back-end functionality, however, our goal is simpler than that. Firebase suits our needs best as we don't need to manage any servers and can retrieve information from our database with ease. The goal of Firebase is to make server eventing simple for devs and thus it fits perfectly within the scope of our project.

3 Prior Work

The skate community as a whole doesn't have any apps that come remotely close to what I've described above. Most skateboarding content and information comes in the form of instructional videos. This makes sense after all, as skateboarding is a complex activity and is best described through visual media rather than a wordy description. Looking up ”How To...” followed by some skateboarding trick will give you a ton of videos geared towards beginners with step by step walk-through. These videos are fantastic and super helpful for anyone approaching the sport and looking for a way to get started.

I plan on binge watching a significant portion of these videos when creating my app and it will serve as a fantastic reference guide to understand the ins and outs of more complex tricks and an excellent basis for most of the terms I plan on defining. However, when it comes to written content, this is where skateboarding really falls short. There are few written guides, those that do exist are generally poorly organized making them not ideal for newcomers, poorly written and full of mid 2000's internet meme flair, or midway through attempt to sell you some new and revolutionary skate tool.

The most notable piece of prior work is that of the FGC Glossary [3]. The FGC Glossary is a fantastic website which was designed from the ground up to deliver digestible information to newcomers to the fighting game community. This website is where I plan on taking most of my inspiration for my app from as it has near identical design goals and features. Each term is defined succinctly, has a video player which can be expanded to show a live example, and has heavy inter-linking between terms and concepts in a series of nested lists. On top of that, one is able to link directly to and from terms as a means to easily share it with someone else. The Fighting Game Community has a lot of the same problems as the skateboarding community; many terms or ideas require prerequisite knowledge that beginners don't have. Instead of asking if they know what an ‘Attack Cancel’ is, you can just send them the link to the definition of a ‘Kara Cancel’ and they can click any of the links to answer any of their questions.



Black Beat Combo

The Guilty Gear-specific term for an invalid combo. This means that the timing of your air combo was not precise enough, and you left a chance for your opponent to air tech. The combo counter goes dark (hence the name Black Beat) to indicate that the combo was escapable, but it's up to your opponent to be on the ball here. In some versions of Guilty Gear, they will even tell you which hit of the combo was the fail point by leaving a number under the combo counter. In BlazBlue, this situation is called a Blue Beat Combo, since the combo counter turns blue instead of black, but the idea is the same.

黒ビート (kuro bīto) – *Lit.* black beat (in Guilty Gear)
青ビート (ao bīto) – *Lit.* blue beat (in BlazBlue)

Air Tech

Recovering from being hit in the air and returning to neutral. Maybe the best way to think about it is like quick rising in the air once you've run out of hit stun. Air teching is usually only in games with a high focus on aerial mobility, like anime games or team games. It's almost always shortened to just "tech", which makes it overlap with (several!) other terms, but you can usually figure out what people mean based on context.

空中受け身 (kūchū ukemi) – *Lit.* air receiving body

Invalid Combo

A "combo" in some anime games that the opponent could have air teched out of. This happens when you mess up the timing of your air combo, but the opponent does not correctly air tech when they were able, so you got to continue hitting them anyway. Invalid Combo is the general term, but some communities will have a name specific to their game, and it's usually related to how the color of the combo counter changes when an invalid combo happens. For example, Dragon Ball FighterZ players will call it a "Blue Combo" and the Guilty Gear community calls it a Black Beat Combo.

Anime Game

A particular style of fighting game that often employs frantic, highly aerial-based combat (including air dashing) and wild character designs, often drawn with a Japanese anime aesthetic. There are many popular anime fighting games on the market, from original properties like Guilty Gear or Under Night In-Birth, to famous licensed IPs like Dragon Ball

The FGC Glossary's means of displaying information and linking between terms is ideal due to its simple nature that gives powerful tools to its user. One can understand how it works at a glance which makes it great for beginners.

4 Methods

I'm going to begin development by creating a simple mock-up in Figma. Figma is a powerful design and visualization tool run in the browser that allows for easy wireframe mock-ups to create the absolute basics of an app. Once I've found the mock-up to be suitable I'll then begin development in React. React has ample documentation online which makes it really easy to get up and running. Before I begin to focus too much on visuals and UI, I first need to make sure that my idea of creating nested lists can be accomplished. While I'm confident the look and design I'm going for can be done in React, both NodeJS and React Native have a vibrant ecosystem of extensions and add-ons that ensure what I'm envisioning can be accomplished. The interface by its design is going to be extremely simple in order to make it as accessible as possible. From there I need to begin development on creating the database of tricks and skate concepts. This is where most of my time will have to be devoted. Not only will I need to research and write perhaps a hundred or more descriptions about various skateboarding moves, I also need to make sure each one properly links to the other, and can be easily updated in the future. I plan on reading and watching as much skate information as I can in order to ensure the information I'm writing down is accurate. Once I have a product that works just barely I plan on shipping it out to anyone and everyone who skates. Getting their feedback early on in the design process is important as the whole goal of the app is to specifically cater to them.

5 Evaluation

The Evaluation part of the app is going to be difficult. It's a hard task to quantify if your knowledge collection reference book app is actually any good. With that being said, Occidental is the perfect place to get information. On campus there are tons and tons of skateboarders from all different backgrounds and skill levels, many of which I know personally and could leverage in order to receive feedback on the app itself. Although the problem surrounding how to quantify the feedback still remains, the community present at this school will be a massive boon to the project.

The simplest solution would be to devise a questionnaire for people to fill out after using the app for a period of time and if they enjoyed using it, what they liked about it, etc. The list of questions would have to be exhaustive but at the same time remain short enough that it doesn't become a chore to fill out and provide feedback.

One possible problem with this solution would be how I know most of the skateboarders personally, and their feedback to me would be swayed because of our personal relationship. It's no longer feedback based on if the app is good or bad, but telling Brady if his app was good or bad. This could be alleviated somewhat by implementing an anonymous feedback form but much of the problem of my personal relationship to these skateboarders still remains.

Perhaps the truest test to see if the app was actually good or not was if people actually used it. A reference manual is only helpful if it's actually opened and its information absorbed. It could have the most beautifully designed UI ever and the most well-thought out organizational structure you can imagine, but if people don't want to check the app for information then it's all worthless. I also don't want to tell people to open the app at least X amount of times as that also defeats the purpose of the evaluation. People should naturally be drawn to it whenever they have a question that needs answering or a refresher on a specific trick. The evaluation part of the app definitely needs to be expanded upon, however, I think currently the best solution is to give people access to it over the course of the month. At the end of the month inquire about a short survey for them to fill out, and then cross check the answers they've given on the survey with in-app analytics. This would offset the potential biased answers and also provide insightful information by seeing if people only opened it up once and never again, or on a semi-regular basis whenever they had a question. These analytics could also guide potential future development and uncover potential areas that people need help with when skateboarding.

6 Ethical Considerations

Ethical considerations must be made, especially in consideration to the sourcing of definitions and images. A key area of contention within the skateboarding community is what defines a specific trick. Due to the decentralized and grass roots nature of what each trick is called, there is a high degree of separation and lack of uniform terms. Someone from the East Coast might call it a '360 flip' while someone from the West Coast might call it a 'TreFlip'. Some believe an 'Impossible Flip' must flip around the front foot and use it as its axle of rotation, others believe the board just has to complete a full flip and its relation to your foot is inconsequential. The dilemma in this case is who is right, the ethical dilemma is who gets to decide who's right? I am by no means an arbiter for skateboard terminology nor an expert by any means, because of that it seems ethically dishonest if I'm the one who decides what gets called what. To offset this I've decided to select and source definitions to the best of my ability, from there I provide similar aliases that redirect to the same definition in case there is a regional disparity between terms. Lastly, and the most important part, on each page is a link to dispute, add, or totally change a definition. While this will require some significant back-end infrastructure, it's important to make sure the community has the option to voice their concerns or clarification.

7 Timeline

- Summer 2022:

  Research and teach myself the best practice for developing native apps on both iOS and Android. This includes but isn't limited to: JavaScript design, principles of React architecture, and Firebase design and security. Create initial wire-frame mock-up in Figma. During this time I also plan on watching and taking notes on a host of different skateboard videos, ensuring that when I'm finally ready to import them into a database I have all the information necessary to do so.

  By the time school starts again in the late summer I plan on having a functional app that I can begin to test. Definitions at this point would be scarce but I plan on making sure the systems are in place so I can rapidly add to it.

- Fall 2022:

  When we arrive back on campus I plan on immediately sharing the minimum value product to everyone in order to get ample feedback. By having the initial prototype done so soon it will allow me to seek feedback and constantly reiterate upon my design.

- Winter 2022:

  The app at this point would have been iterated upon for at least a couple months meaning I would be close to the end goal of having it be fully featured and near exhaustive when it comes to definitions. At this point in time the app would have been in testing for long enough so I could begin to leverage the in-app analytics to tweak and change the design accordingly.

References

[1] Bush, Vannevar. *As We May Think*. July, 1945.

[2] Chernev, Alexander, Böckenholt, Ulf, and Goodman, Joseph. "Choice overload: A conceptual review and meta-analysis". In: *Journal of Consumer Psychology* 25.2 (2015), pp. 333–358. ISSN: 1057-7408. DOI: https://doi.org/10.1016/j.jcps.2014.08.002. URL: https://www.sciencedirect.com/science/article/pii/S1057740814000916.

[3] Infil. *The FGC Glossary*. 2020. URL: https://glossary.infil.net/.

[4] Luhmann, Niklas. *Communicating with Slip Boxes*. 1992.

[5] Nzskater4life. *Hospital Flip*. 2009. URL: https://skateboarding.fandom.com/wiki/Hospital_flip.

[6] TonyHawk. 2004. URL: https://quotefancy.com/quote/1229413/Tony-Hawk-I-consider-skateboarding-an-art-form-a-lifestyle-and-a-sport-Action-sport-would.

[7] Weyland, Jocko. *The Answer Is Never: A Skateboarder's History of the world*. Grove Press, 2002.