Behavioral Change Therapy Practices and Habit Building

Abstract

While many websites do already exist in the self-help and habit-building space, many do not apply principles of behavioral interventions to increase the effectiveness and likelihood that those behaviors will become habits or that the user will achieve whatever goals they have set for themselves. As such, my senior project was to design and implement a website that allows users to choose certain behaviors, set goals for themselves, and then track their behavior over time. Built into the web application are several different functionalities that look at the antecedents of the behavior as well as the consequences. Overall, the principles of behavioral change therapy focused around peer involvement, prompts, feedback and reinforcement.

Background and Objectives

There is a lot of “unmet need” in the space of mental health in the US. Approximately one quarter of the entire population suffers from at least one psychiatric disorder and the large majority receives no help. [1, 2] That means almost one in every five people in the US suffers from some type of mental health disorder and receives no treatment for it. The interest in self-administered psychological interventions has been great because of this, and because of a myriad of other reasons including its possible wide applications and ability to affect public health [3].

This is clear when we look at what applications this web app may have in the real world. While we are building habits like exercise and meditation, the platform could just as easily be used to track things like calories eaten, the number of alcoholic drinks consumed, or the number of cigarettes smoked. Goals could then be set to lower these measures and reinforcements could be given accordingly. If verification was added to the scores, it may even make sense one day for health insurance companies to use a product like this to reduce smoking and drinking, preferring to prevent the unhealthy consequences of these behaviors rather than pay for them. Developing a platform that increases self-control and gives people a way to address their own problems is powerful at a personal level, as well as at a public health one.
This intersection of psychology and computer science is one that interests me greatly, particularly as the number of novel psychological treatments that use some form of technology has been increasing dramatically. Whether those treatments involve the internet or mobile devices, it’s indicative of a willingness to try new things and to use whatever tools are available. As such, I saw my senior project as an opportunity to build out a tool that might one day be expanded into a novel self-administered treatment.

While there also may be certain ethical concerns with data collection, I do think that a web application such as this provides a wealth of data and a platform to test different reinforcement strategies and behavioral interventions. The idea of AB testing could be easily applied to test two different popular interventions (users randomly assigned to either one). What could then be measured are things like the percent of users who achieve their goals, the number of posts per person, etc. ABAB testing could also be used. For example, during certain phases, no reminder emails are sent and no reinforcements given out to see whether or not emails serve as effective prompts or whether the reinforcements being offered are strong enough to affect behavior.

**Technologies Implemented**

The website was created using the Ruby on Rails web application framework and used a combination of HTML, CSS and JavaScript to achieve its effects. Ruby on Rails was chosen for its ability to quickly write a web application as I have never done web development before. I also have several friends who have coded in Rails, and the ability to ask them for help as I was getting started was another consideration that pushed me towards Rails over PHP, Python and Django.

Twitter Bootstrap, a front-end framework that contains certain HTML and CSS design elements, was also used. The use of bootstrap was primarily to utilize its support of responsive design, but it was also chosen for the large library of free bootstrap tutorials and themes on the Internet. A PostgreSQL database was used to store all user, goal, phase and record data. Github was used for version control and the application is hosted on Heroku.

**Description of Database Schema/Active Record Associations**

The full schema is available in the source code under db/schema.rb, but this section will summarize the structure of my database, and the reasoning behind it. The first component of my web application was the “user” model. There is nothing
particularly interesting of note here, but users do have roles, where certain users are created as “admins” and have the ability to manage other users. The base of my web application was based on a tutorial by Daniel Kehoe that used devise, a Rails gem for user management and authentication, and another Rails gem, Pundit, for authorization. The tutorial also used Bootstrap as the front-end framework, and gave me a solid foundation from which to work.

I then built the relationship between users and goals, where users have the ability to add many “goals,” of which there are six types (exercising, journaling, running, meditating, doing yoga and drinking water). Because much of the goal functionality is shared across goal models, I wanted to keep the code as DRY as possible, and each of the goal types extends from the original goal model.

Each goal then has many phases, which sets a baseline score of 0 if the behavior is new, or lets the user set baseline scores if they are already performing the action. I decided to include another layer of depth here so that behaviors could be broken up into weeks (there are seven records in a phase), and so that sub-goals could be set. A changing criterion design (CCD) was used as the primary intervention design, and phases mirror the structure used in CCDs well.

Phases are never created by the user, but are automatically created when they reach seven records. Records are the model that keeps track of the user’s daily score, while phases are used to set sub-goals for the user to strive towards. This breakdown of phases and records also allows for the means of records to be more easily calculated and the slopes plotted. As each user is his or her own single-case design, this also allows for the data to be more easily evaluated under the four visual inspection criteria (changes in means, changes in slop, shifts in level and latency of changes) [3].

User Experience

The first time that you visit the website, you are shown the landing page, which is a responsive bootstrap theme that includes links in the navigation bar to login and sign-up, as well as a short description of what the website does and links to the goals pages. Below is a screenshot as it would be viewed on a desktop. The theme for the landing page was a free template from http://startbootstrap.com/. While some changes were made to the navbar and the div towards the bottom, the image and CSS for the page was left relatively unchanged.
If the page is viewed on other devices, the layout automatically reconfigures to a different column designs (this is also true if the window on a desktop is not maximized). Below are two screenshots of this. Other pages will not be included for both formats but I did want to include these to showcase the benefit of using Twitter Bootstrap and the responsive nature of the website. This makes viewing on mobile devices easier and while the project was done as a web application and not a mobile phone application, there is not limited access on mobile devices.
Everyone Has Goals

We are here to help you achieve them. Using well-researched principles of behavior change therapy, we're using psychology to help you accomplish the things you want to accomplish.

SET A GOAL TODAY

How We Help

Cooperation

Setting goals with friends is helpful for both parties. Not only can you push each other on a daily basis, but now, someone knows if you didn't follow through. Using Metamorphosis with friends is also a great way to increase motivation and greatly increase the likelihood both parties reach their goals.

Reinforcement

Reinforcing a behavior is key to increasing the likelihood of performing it again. If you want to put your money where your mouth is, add a monetary value to your goal. Every day you get closer to achieving your goal, you get a corresponding amount of your money back. When you reach it, we’ll refund the charge.

Tracking

Other websites only track whether or not you performed an action (binary), but research has shown that tracking your progress over time is in itself a self-reinforcing behavior. We make it easy to see where you started from, how far you’ve come, and how the task has gotten easier over time.

SET A GOAL TODAY

Cooperation

Setting goals with friends is helpful for both parties. Not only can you push each other on a daily basis, but now, someone knows if you didn’t follow through. Using Metamorphosis with friends is also a great way to increase motivation and greatly increase the likelihood both parties reach their goals.

Reinforcement

Reinforcing a behavior is key to increasing the likelihood of performing it again. If you want to put your money where your mouth is, add a monetary value to your goal. Every day you get closer to achieving your goal, you get a corresponding amount of your money back. When you reach it, we’ll refund the charge.

Tracking

Other websites only track whether or not you performed an action (binary), but research has shown that tracking your progress over time is in itself a self-reinforcing behavior. We make it easy to see where you started from, how far you’ve come, and how the task has gotten easier over time.
You can see the types of goals that can be set without logging in, but all goal functionality is only available after an account is created and a user logged in. Email addresses must be confirmed prior to use. Below is the page that lists the six types of goals and gives you the ability to add goals.

Choose Your Goal

- **Run**: Humans were born to run, and running is one of the oldest forms of cardiovascular exercise. Running is a great way to get in shape. There's no cost, besides a pair of shoes, and you can run almost anywhere.
- **Drink More Water**: Every day, you're supposed to drink eight glasses of water. If you exercise, maybe even more. Staying hydrated has an abundance of healthy benefits, but it's hard to remember to keep drinking water.
- **Do Yoga**: Those who do yoga swear by it, and for good reason. This ancient art has been practiced for thousands of years as it increases flexibility, offers a core workout and is a physical, mental and spiritual practice.
- **Meditate**: Meditation, like yoga, has a multitude of physical and mental health benefits. Some of the reasons that meditation may be practiced are to increase physical relaxation, to decrease anxiety and to cope with stress.
- **Journal**: There is ever increasing evidence that journaling can affect wellbeing, and journaling can be useful for some people to deal with stressful events. All you need is a pen and paper, so why not start today?
- **Exercise**: The benefits of exercise are well known and supported, and the influx of gym memberships at the beginning of every year suggest that people want to exercise more. Let us help you build the habit.

Clicking on any of the “Start Goal Behavior” buttons leads you to the new goal form, which is included below. For running, doing yoga, meditating, journaling and exercising, the goals are set on a minute/day basis, while for drinking water, the goal is set on an ounces/day basis. All forms are similar though, with the ability to choose different goal targets from a dropdown menu.
Once a goal is created, the structure of the goal page depends on whether or not the behavior is new. If the new behavior checkbox is checked, then we know that the user is not already performing the action and as such, a baseline phase with seven records of score zero are automatically created to prepopulate the chart and to draw more attention to the growth once they do start. Screenshots of both are included below.
Both options allow users to make a monetary commitment if they would like. I have arbitrarily set the commitment to $50, but the basic idea behind the commitment is that users are charged $50 if they agree to it, and then when users reach their goal, the full charge will be refunded. This allows for us to use money as a reinforcer, as you now “earn” some percent of your commitment back for doing the correct behavior. If you input a score, regardless of what it is, you will earn half of a percent of your commitment back ($0.25). If your score reaches your subgoal (initially set at zero for baseline behaviors, five minutes for minute-based goals and 36-ounces for the increased water consumption goal), then you will earn another half a percent.

The reason I have the reinforcements for both is because I want to reinforce submitting scores and doing the behavior on a daily basis as much as I want to reinforce reaching the goal. Because of this, you also gain 1% of your commitment back for every three days of consecutive input and 5% at the beginning of a phase. The goal of the monetary commitment is to act as a strong reinforcer when the correct behaviors (inputting scores, reaching sub-goals and goals) are seen. A progress bar tracks your progress toward earning your full commitment back.

I have also built feedback into the graph, as the bar will be green when you reached your subgoal for the day, yellow if you reached 80% of your subgoal or red if you did not reach either. Below is a screenshot that shows this.
The blue lines on the graph indicate the sub-goal for the phase, and as you can see, because the average score for the first phase was higher than the sub-goal for that phase, the sub-goal increased for the following phase (moving from five to ten). Daily records do not only include scores but also an easiness rating, which can be graphed on top of the record data so that the user can see whether or not performing the behavior gets easier over time, even as the amount of the behavior they do increases.

Because users can have multiple goals, I also included the concept of a goals dashboard. This allows for users to see all of their graphs simultaneously as well as add scores to each of them without having to navigate to the individual goal page. The root of the website defaults to the dashboard when the user has added at least one goal, while it remains the landing page up until that point. The dashboard also includes an activity feed, which shows all of the user-generated record data. The activity feed will include all records submitted by the user, as well as all records for users that the current user is following. Similar to Twitter, we then have the ability to subscribe to other users, in which case their goal data will populate our news feed. This is important because it allows friends to see how each other are doing and follow along with each other’s progress. It also serves as a social reinforcer, as each item on the feed says whether the goal was met, not met or surpassed. Peer involvement in behavioral interventions has been shown in certain situations to enhance performance [3] so adding some element of cooperation was important.
As I thought about my app, I wanted some way to increase the likelihood of performing an action that did not focus on the consequences of the behavior (reinforcement, feedback, social pressure, etc.) As such, I decided to look at the antecedents to the behavior as well, and came up with the idea of sending a reminder e-mail as a prompt. My logic was that sending an e-mail as reinforcement did not make much sense because getting an e-mail is not necessarily reinforcing in any way—if anything, with the large number of junk emails we sift through on a daily basis, it may be seen as a punishment. As such, I decided to send e-mails as prompts (23 hours after a record was created). The purpose of delaying the e-mail is to have the behavior occur at the same time and place each day, making it more likely that it will become a habit. It also makes the assumption that if you were free yesterday at a given time, there is a higher likelihood that you will be free today as well. If the user is always in the same place around the same time, the environment itself may become a setting event for the behavior which makes the likelihood of the behavior occurring even higher.

**Difficulties Encountered**

As I have previously done very little web development, I had to learn Ruby on Rails, HTML, CSS and JavaScript, as well as basic SQL queries to finish my senior project. This meant that I also had to familiarize myself with many Rails conventions, like the model-view-controller structure that Rails runs on, as well as how to set up routes and the relationship between web applications and the Internet as a whole (http requests, for example). Coding in Ruby itself was definitely a lot faster than coding in C, which is the only other language that I have experience with, but understanding how web applications work did take me quite some time.

Initially, I also ran into several problems setting up the development environment on my Windows computer and eventually ended up coding on a MacBook. While there is a lot of documentation on coding in Rails on Windows computers, much of the troubleshooting is for older versions of Rails (Rails 3 vs. Rails 4) and many small problems took me days to fix (setting up the command prompt, RVM, etc.). I did run into several problems setting up the development environment on my MacBook, as some similar problems popped up regarding the version of OSX that I was running, but updating to Mavericks helped a lot and there were very thorough guides explaining how to install Ruby, Rails, RVM, etc.

My next biggest difficulty came with setting up the goal architecture. I did not want to recreate the code in each of my goal models as it seemed like the behavior should be inherited from a single goal model, but understanding how to set up the routes and use the controller correctly was difficult. I did, however, end up having all of my
code in one goal model and one goal controller, with my different goal-types existing in empty models. I’ve included one such model below to illustrate this point.

```ruby
Exercise.rb
Class Exercise < Goal end
```

After setting up the relationship between users and goals and understanding how to build the goal through the user to automatically set the user_id, I thought that setting up phases and records would then be relatively easy. I did not, however, want to build phases and records in a similar way, and felt that phases should be automatically set with its own logic, and records should be added from the goal controller and view. While many Rails apps have one or two levels of this sort of parent/child relationship, I had quite a few more and understanding exactly how to structure my Rails views, routes and controllers was quite difficult as well.

For the most part, these were the issues that I ran into that involved Rails itself and the structure of my Rails application. The other major difficulties I encountered had to do with specific gems and integrating other components of my website. Now that I had very basic tables set up and the ability to create them as I wanted, I began to work on other aspects of the user experience (such as the JavaScript charts).

Working with amCharts provided its own challenges. After finally getting a decent grip on how Rails worked, I was back at square one when it came to JavaScript. While they have many examples of the different kinds of charts you can create, integrating the charts into Rails and understanding how to create the chart that I had envisioned was much more challenging. While I could create a line graph quite easily, understanding how to combine line graphs and bar graphs was harder. I also struggled to convert the right Rails information into data that the charts could read, and getting the logic of different colored bars to work out took some time as well. I ended up having to add several other attributes to my models to get the charts to work (such as a color and sub_goal attributes for my records model).

My struggles with JavaScript continued as I tried to set up the Stripe payment to allow for “commitment contracts.” It took me about five minutes to get the pay button to successfully charge cards, but it took me much longer to understand how to feed that data back into my Rails app so that money could be refunded and how to change my model attributes to hold onto the card data.

My final major struggle involved SQL queries. While Michael Hartl has a great tutorial on creating a twitter-esq newsfeed for microposts, I struggled to mirror the implementation in my code due to the depth of my relationships. While I could create easily a newsfeed for “goals,” I did not think that a newsfeed for goals made
nearly as much sense as a newsfeed for records would. Yet understanding how to make that query was quite difficult, and even when I did succeed, I realized that it caused significant lag, particularly on my dashboard. To address that issue, I added a user_id attribute onto my records model that inherited the information from the goal so that one query could get the information I needed (rather than four).

**Next Steps and Improvements**

This project offers a good base to move forward from, but there are still several improvements that I would have liked to make given more time. I think integration with Facebook would be a major benefit for the effectiveness of an intervention. Not only does it give you access to the user's social network which makes following their friends and people they are interested in that much easier, but it also provides avenues for reinforcement and punishment. On days when the user does not reach their goal behavior, it could post onto their wall about the failure, which is annoying for the user and his or her friends and embarrassing for the user. Similarly, on days when the user beats their goal handedly, it could post success on their wall as well and this would allow their friends to reinforce the behavior with likes. Even without Facebook integration, an ability to “like” activity would definitely make peer interactions more powerful as it makes it clear that others are watching and gives peers an avenue to reinforce good behavior.

Another improvement that could be made is the ability to add “group” goals, where a group of friends can work towards a daily goal that sums up each person’s contribution. This type of group-based treatment has been shown to be effective in certain cases, and it also lends itself to team-based treatments where competition becomes another motivation for performing the action [3]. Then, when a user does not reach their daily sub-goal, they are not the only ones who are affected.

Currently, all of the behavior is being self-reported so the validity of the measurements is somewhat questionable. Building in a mechanism where users can have their scores validated by a second user or some other check of interobserver agreement would make the results of any kind of empirical comparison between interventions more meaningful.

Other types of behavior interventions could also be coded (using different prompts, setting events, feedback, reinforcement, etc.) and compared to the project above and even with just a simple binary calendar (each day, you check if you did something).

**Notes**

2. Github repository at [https://github.com/kzli216/seniorproject](https://github.com/kzli216/seniorproject)

References