

# Team 10

## Engineering

## Accountability Log

Michael Lipski  
Kevin McGuire  
Thomas Neter  
Greg Hoerdt  
Peter Marshall  
Griffin Cook

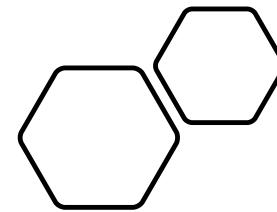


# Log Legend

The colours on the left side of our slides represent the slide's type of entry. Sometimes there is slight overlap, in which case the most relevant category is highlighted.

- **Blue** - Meeting/Planning
- **Green** – Research
- **Red** - Prototype
- **Purple** – User Testing

# Pre Sprint

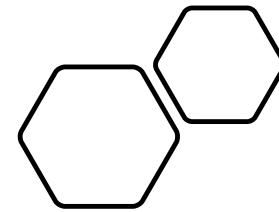


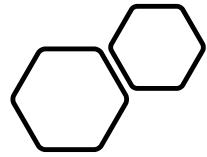
May 22, 2020  
Everyone  
Notes by Griffin

# First Meeting

- Team video chat.
- Greg and Peter both have musical backgrounds.
- Greg mentioned he thinks using GA's to get chord progressions won't be too bad.
- We all agree that the networking part and optimizing data transfer times will likely be the hardest issue.
- We need to figure out what kind of software we want to use to make the user interface assuming it doesn't have to be in Processing.

# Sprint





# "Monday"

Sprint Questions, HMWs, and Map

May 28, 2020  
Griffin

# Persona – Dave Josephson

## **Who:**

- 27 years old
- Always wanted to play guitar but never learned
- Played recorder in grade 4, tuba in grades 7,8
- Doesn't know much about music theory
- Has young children
- Works construction 50 hours a week



## **Motivations:**

- Wants to make music with his friends
- Wants to make music quickly without worrying too much about learning theory
- Secretly wants to become big on Soundcloud

## **Frustrations:**

- Can't play music out loud at night because of young children sleeping
- Always tired from work
- Fingers are big and not nimble

May 28, 2020  
Kevin

# Persona – Belethor

- Local Businessman
- Recently learned about "soundcloud" and wants to record music
- Doesn't have any instruments, but has odds and ends that sort of thing.
- Would like to create music with his friends but they are spread out over the city

## Motivations:

- Make chill lofi beats relax and to pass the time in his store
- Enjoy jamming out with friends, and reduce the anxiety of the crisis
- Not experience lag, hometown has notoriously bad wifi
- 

## Frustrations:

- No one visits his store, and he is bored
- Has no musical instrument skill, or understanding of musical theory
- Cannot sing, or shout as well as some of his neighbors
- The import of new instruments has been banned by the Empire as it has been deemed nonessential



## Possible User groups:

- Creatives
- Professional Musicians
- Amateur Musicians
- Families
- Groups of Friends

May 28, 2020  
Michael

# Personas

User	Relevance to SoC	About
Amateur creator	New to music, wants to create something with this new app or just mess around and find some sick beats	Probably impatient, needs to be drawn in, wants to see results quickly Doesn't know what everything means, needs things explained Might have a friend or two they want to collaborate with but not too serious. Might appreciate a good way to share their work
Expert creator	Is very comfortable with creating music, uses product with an end goal/piece of music in mind	Patient, appreciates the creative process, is willing to learn the UI and see what the app has in store Knows what things are, but still wants to be able to recognize things quickly
Garage band[member]	Open-minded and looking for a way to find inspiration for some new songs through some beats	Patient with the creative process, willing to test out the full functionality, maybe wants to feed in their own music and see what it comes up with Impatient with collaboration, if they feel like this can't replace the in-person experience they will look elsewhere -> need to offer something that in-person can't

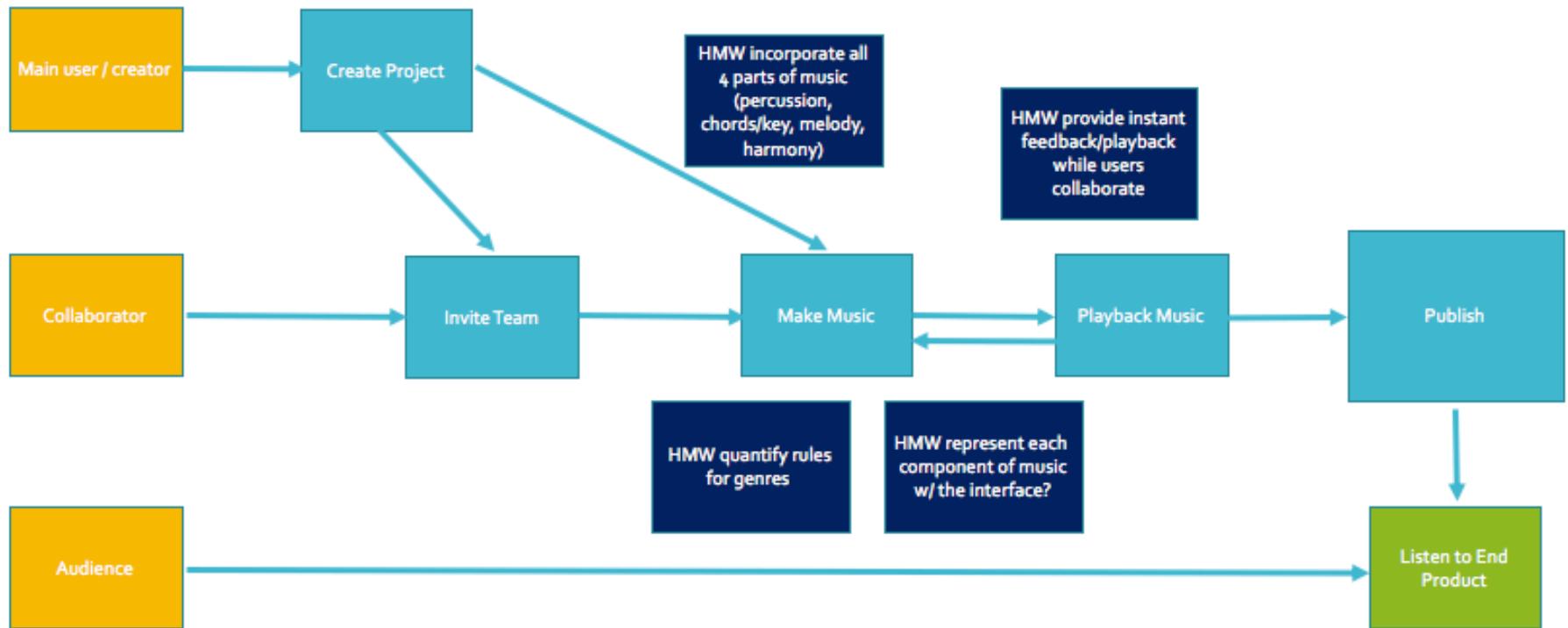
# Key Sprint Questions

After individually writing sprint questions, we collectively agreed upon the following as our key sprint questions:

- *How can we effectively test this product? What are our standards for success?*
- *What are the requirements for successful real-time musical collaboration over a network?*
- *How can we evaluate our creations without bias (user testing framework)?*
- *How much musical knowledge should the user need?*

May 28, 2020  
Everyone

# Map



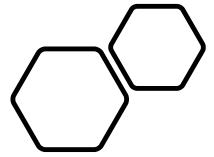
Target User: The main user (could be a novice)  
Target Event: Make Music

May 28, 2020  
Everyone

# Turning the map into "how might we"'s

Heat mappings were used to develop priorities for the first prototype

Collaboration + Network	GA's	Misc	UI/UX (Interface & Community)
HMW provide instant feedback/playback while users collaborate	HMW allow customization within our defined subset of sounds	HMW specify a genre	How might we gamify or incentivize users to keep creating/playing?
HMW create turn-based music	How might we confine the user within a genre in terms of the key, chords, and notes that they are allowed to play?	HMW get our soundbites	HMW lead users to a reasonable result for their chosen genre?
HMW prevent conflict between users' versions of the project	How control the GA based on genre?		How visually distinguish musical components/genre?
HMW make it obvious what other users are doing	HMW incorporate all 4 parts of music	HMW streamline the music making process for those with little time	HMW give users variety/customization of sounds
HMW let users provide feedback/communicate	How define chord requirements so the GA can understand them?	HMW allow users to make significant changes in short amount of time	HMW make a "fun" experience
HMW divide the parts of music among actors	How might we quantify the rules of each music genre in order to generate new music within that genre?	HMW increase audience participation	HMW allow the user to feel like they are contributing to the music instead of watching along
HMW nurture a community environment within our project	HMW set the key	HMW include vocals	HMW make it visually pleasing
HMW allow for quick iterative changes	HMW incorporate chords		How represent each component of music w/ the interface?
HMW notify collaborators of changes	HMW break the box of "genres" while preserving good sounding music		HMW allow user to modify chords and key to change genre
HMW ensure users don't overlap their changes	How represent each component of music w/ the GA?	GYM	How let the creators know how the audience feels about the experience?
HMW save projects to come back to later	HMW allow for a variety of sounds		HMW decide an amount of genres users can select
HMW nurture a community feeling			HMW allow sound importing
HMW have instant feedback			
How control the collaborative experience to avoid networking conflicts and make it feel smooth?			
HMW improve synchronization between musicians			



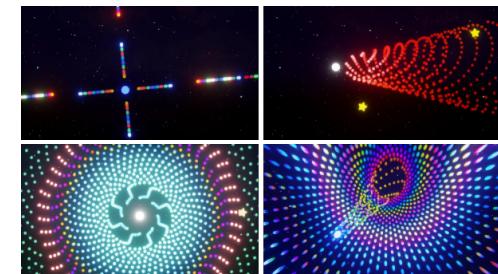
# "Tuesday"

First Designs

# Synesthesia (Lightning Demo) (slide 1/2)

For my lightning demo, I presented an audio-visual music creation game called Synesthesia. The game was created by the Playstation Network user Digitalthing inside of Dreams for PS4 [1]. The player controls a glowing circle that can move around a 2D screen, with the ability to rotate. Each button on the PS4 controller is mapped to a specific musical sound and colour. When the player presses a button, its specific sound is played and its specific colour is emitted in a stream from their circular avatar. The longer the player holds down the button, the longer the sound is sustained and the longer the length of the colourful stream. The player also has the ability to make their circle move fast or slow by toggling a button. When moving quickly, all of the sounds are in a higher pitch, and conversely when moving slowly, the sounds are in a lower pitch. Synesthesia also supports multiplayer, where each player gets their own circular avatar that can create its own sounds and colourful emissions. I presented Synesthesia to my team because I wanted to make us think more abstractly about our user-interface design. I wanted us to escape the idea that a digital music creation interface has to be a bunch of knobs and dials. I also found it interesting how the music was being played in real time, and thought that this could help address our question of “How might we provide instant feedback/playback while users collaborate”.

[1] Digitalthing, "Synesthesia", [indreams.me](https://indreams.me/dream/mZfNaaxdPTH/), 2020. [Online]. Available: <https://indreams.me/dream/mZfNaaxdPTH/>.



May 29, 2020  
Griffin

# Synesthesia (Lightning Demo) (slide 2/2)

One way I thought we could implement something like Synesthesia was by creating a similar platform and using the location of the player to modify an always changing fitness function. The idea was that the user would be able to toggle between musical attributes (ie range of chords, volume), with certain spots describing different manipulations of that attribute (ie. small and large, quiet and loud) located around the plane. The speed the user moved their circle could be used to determine the tempo. The music would constantly be playing, but as the user moved their piece and modified the fitness function, the music would gradually shift to match the changes. The issue with this approach to making a music platform is that it better replicates a jam session more than a recording studio, which does not align with our sprint map. Our map includes a feedback loop between making and listening to the music, allowing for incremental modifications. Our map also ends with the user publishing their song and being able to listen to the final product. In this approach, it could be possible to record the whole session, but the final product would be unpolished, and likely a lot longer than a traditional song.

One aspect of Synesthesia that we all agreed should be implemented in our product is having a visual play alongside the audio. Audio visualization can help people interpret and judge music's structure, and it is a widely used and appreciated concept [2]. For our prototype, we ended up implementing audio visualization by having the notes, chords, and beats of the generated song appear in a grid. When the user plays the music, a highlight effect will move along the columns of notes, chords, and beats, in sync with the audio.

We could take this idea of implementing audio visualization even further in the future by creating a specific listening page within our user-interface. If the users wanted to listen to what they created, they could navigate to this page, where the sole purpose would be to listen to the music without the visual distractions from the controls of the music creation interface. The listening page could be predominantly a visual effect used to enhance the auditory experience. The hypnotizing types of effects used in Synesthesia could fit in well for this use case, further drawing the user into the experience. This page could also be used as a resource that users can share to show off what they have created. It's akin to an abstract music video and it could help entice others to stay on the page longer in a world of saturated media and short attention spans [3].

[2] Y. Zhang, Y. Pan and J. Zhou, "Study on Application of Audio Visualization in New Media Art", Journal of Physics: Conference Series, vol. 1098, 2018. Available: 10.1088/1742-6596/1098/1/012003

[3] H. Griffey, "The lost art of concentration: being distracted in a digital world", The Guardian, 2018. [Online]. Available: <https://www.theguardian.com/lifeandstyle/2018/oct/14/the-lost-art-of-concentration-being-distracted-in-a-digital-world>.

# Sound Trap(Lightning Demo)

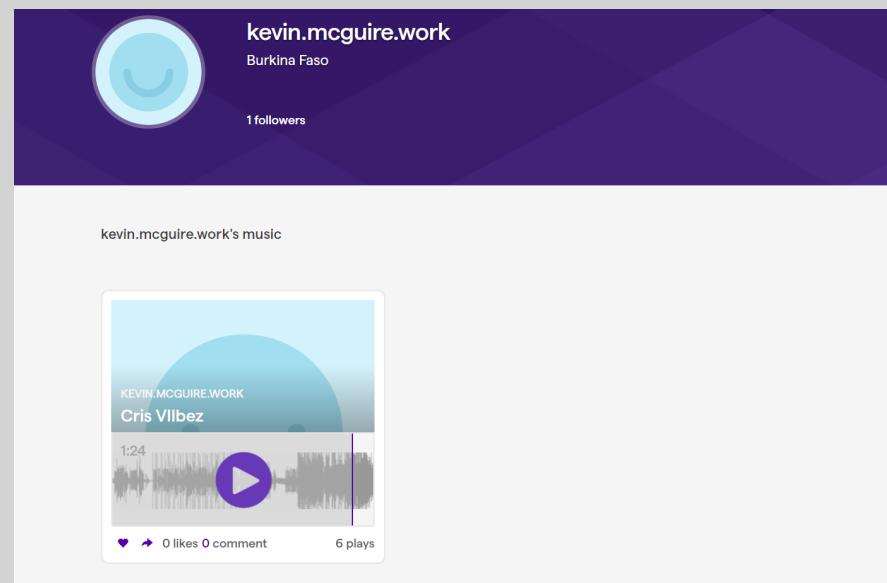
May 29, 2020  
Kevin

What is it?

- A platform that is like garage band you can use online

What I took away from it:

- The UI is unintuitive, and hard to use if you haven't used similar software before
- The collaboration functionality is jerky and not great to use



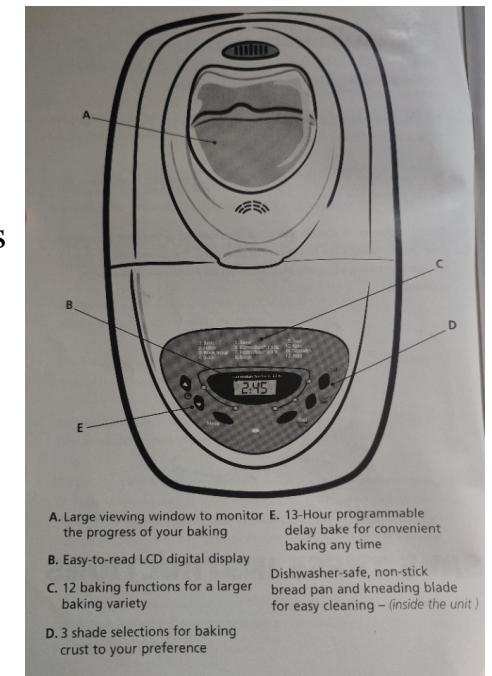
May 29, 2020  
Michael

# Bread maker (Lightning Demo)

(Making bread was demoed)

Referring to the figure to the right, the bread maker comes with an instruction manual explaining how to operate it, but also comes with easy-to-follow instructions for each of the bread types that it supports. A user with little break-making experience can input a few ingredients for a particular type of bread as recommended by the user manual, tell the bread maker the type of bread they are trying to make, and then change a few settings to customize the process based on their preferences for different loaf sizes or crust shades. The user is then rewarded with a delicious end-product that is comparable to professional baking results after waiting for some time.

The GA music making process should ideally be this simple and supportive for the user.



May 29, 2020  
Griffin

# Gather, Doodle and Crazy 8

### Gather

- Seamless collaboration
- HMW handle genre
- HMW represent everything in the UI?
- HMW incorporate percussion chords melody harmony
- Rock
- Pop
- Techno
- Communication
- Fitness - static, dynamic?
- Your Turn
- Play Watch
- Musician vs Professional
- Make Music vs Play Music / Jam
- Adobe
- Basic click menu advanced
- Audio-visual Sound trap brush

### Doodle

"High" blue in orange  
"Low"

Specific shapes for users  
Location & speed  
fitness function

Colour indicates component being modified  
Music constantly plays

Jam session

Time 230s

tempo ↑	genre ↓
0 0 0 0	template ↓
0 0 0 0	roles ↓

- VOICE & text chat
- features are laid out in one UI
- users take turn making their adjustments
- option for time limit per user

Record

Sound 1 priority  
Sound 2 priority  
Record New Sound

each user locks onto a feature they want to modify

input parameters

- other users have temporary viewer privileges to your locked section
- integrated voice & text chats
- ability to turn on "advanced" mode
- everyone has to click publish
- play button plays for you or everyone

### Crazy 8's

Turn base

0 0 0 0	0 0 0 0
0 0 0 0	0 0 0 0
0 0 0 0	0 0 0 0

Roles: musician at record

Tempo

Draw + interpret shapes

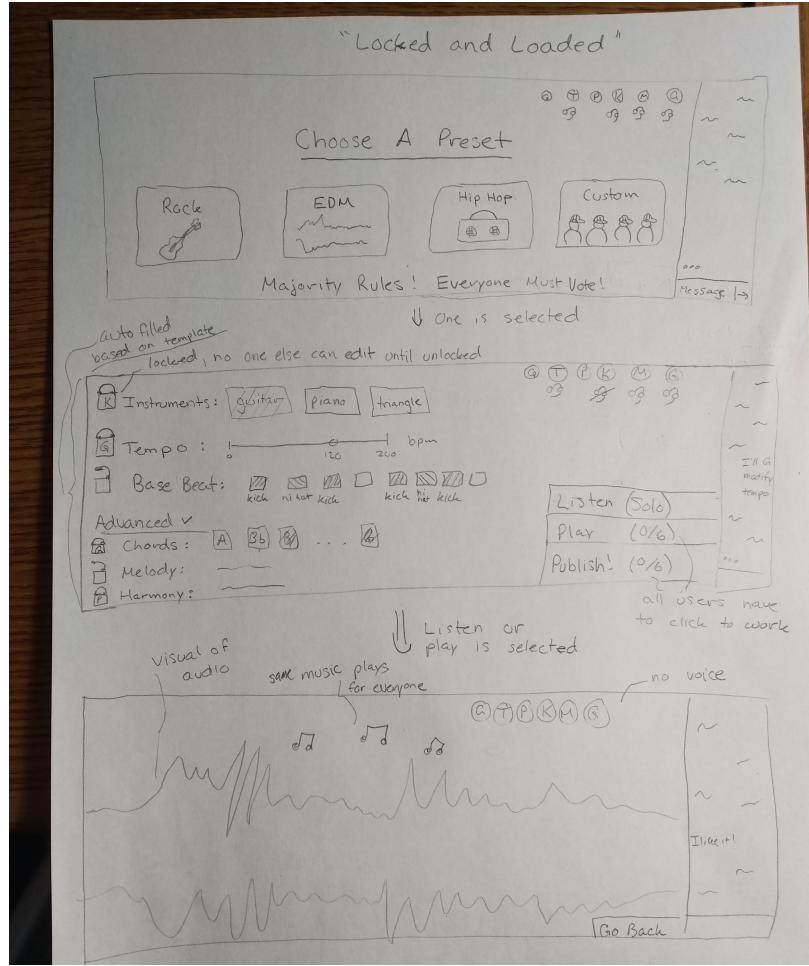
Create Fitness

ability to add vocal on song

interpret image

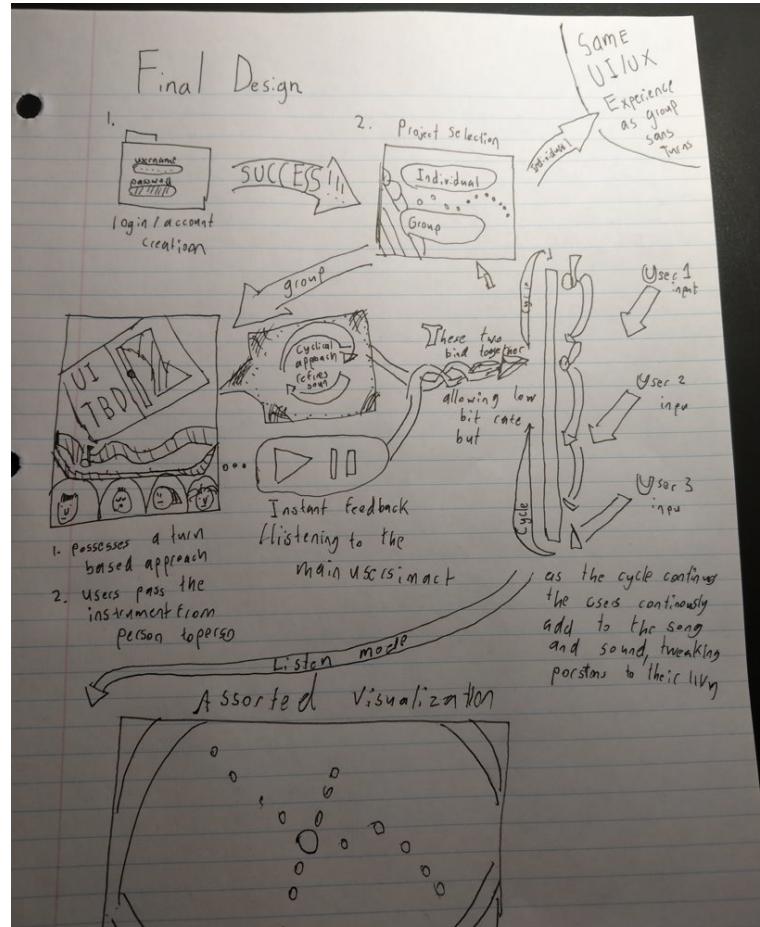
# Design Proposal – Locked and Loaded

May 29, 2020  
Griffin



May 29, 2020  
Kevin

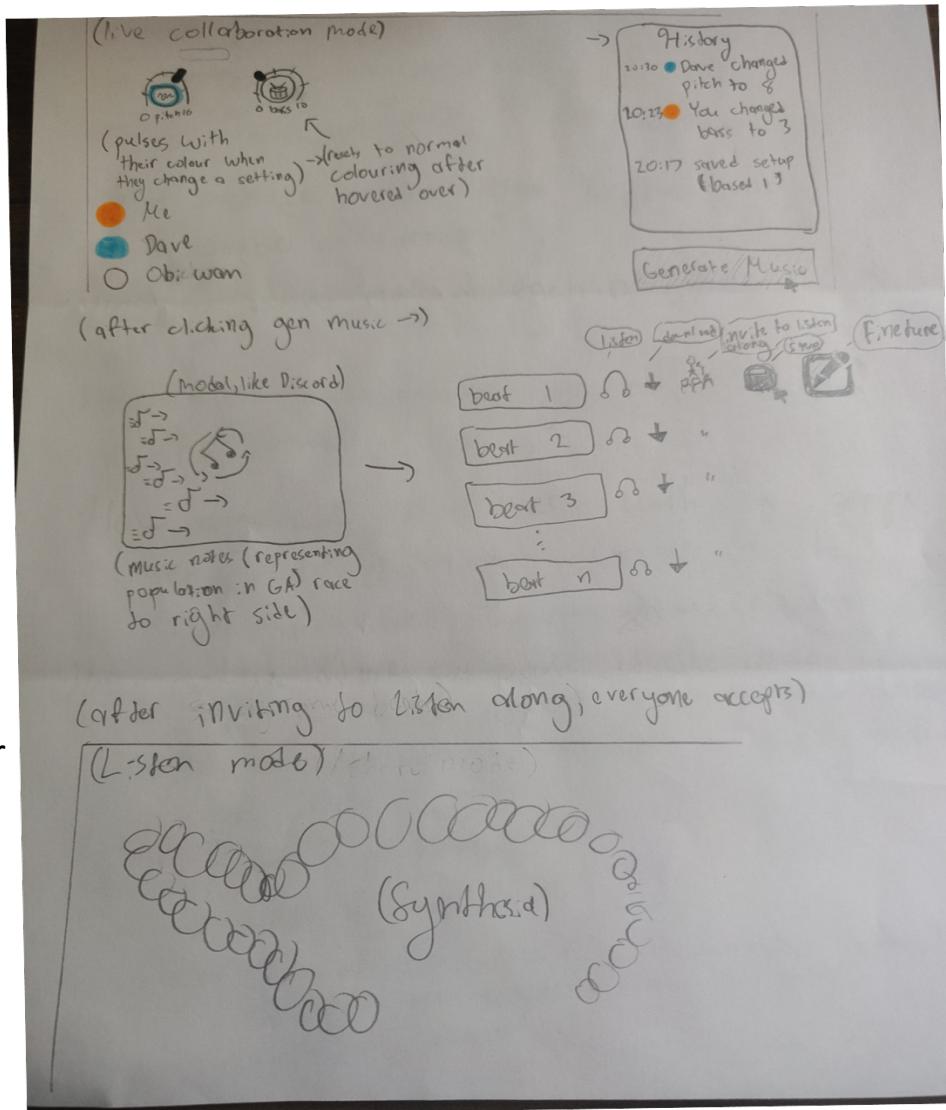
# Design Proposal – Turntable



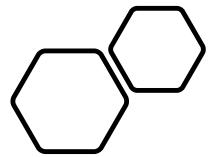
# Design Proposal – Music note pod racing

Notable features:

- Generating music loading modal
- List output of beats as opposed to just 1
- Actions for each output beat (e.g. listen, group listen, finetune, save, etc.)
- History list component
- Who's online list – unique colour for each user
- Change notifications (control pulses with user colour)
- Listen mode experience



May 29, 2020  
Michael

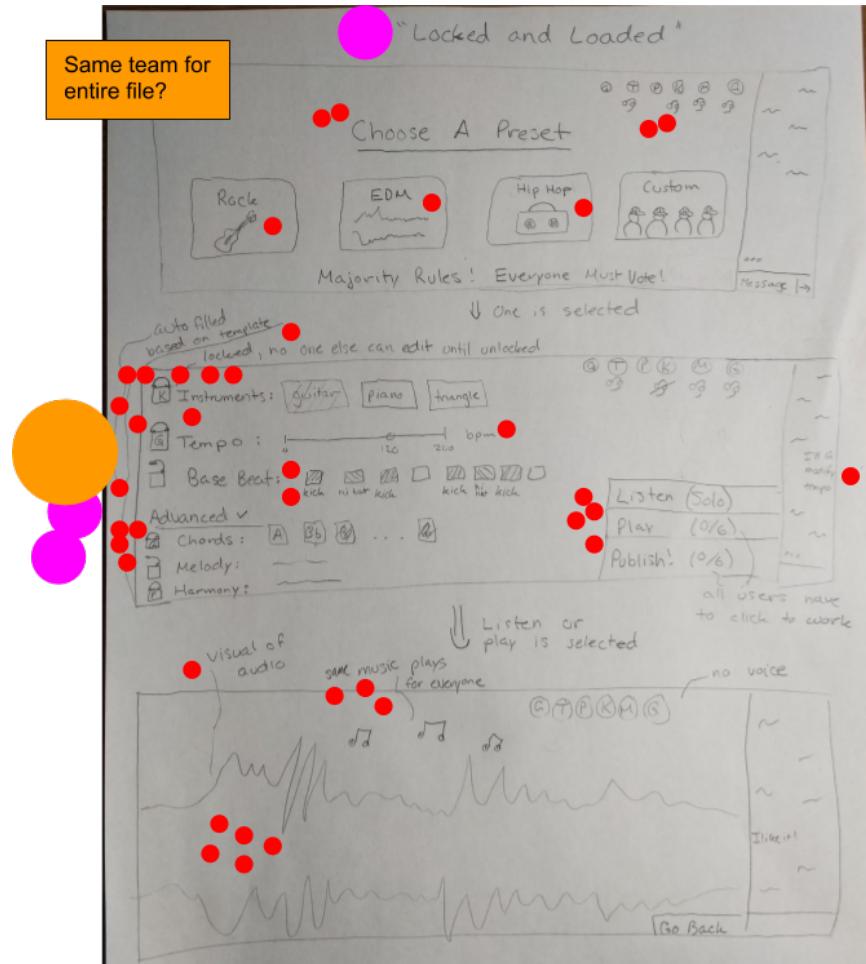


# "Wednesday"

Design Critiques and Storyboard

June 2, 2020  
Everyone

# Design Analysis – Locked and Loaded



## Scribe:

- Users online with muted mics
- Choose a preset genre, with voting
- Basic and advanced configs for inputs to GA
- Separate UI for listening solo or group
- Visualization of sound, frequency of music
- Users lock onto settings



June 2, 2020  
Griffin

# Deciding Designs to Keep

- I was selected as the decider.
- I gave a super vote to each of the following three design elements:
  - Guh's project setup, which included selecting a genre, and having a share link.
  - Tablet's separation of percussion, harmony, chords, and melody.
  - Locked and Loaded's locking system for collaboration.

June 2, 2020  
Everyone

# Product Name

- We all produced potential names.
- Options included Big Brain Beats, Brainless Beats, Robusic, Wetunes, BitBeatz, Looped, and GeneticBeats.
- Using dot voting, we chose **BitBeatz!**

# Story Board

We all created a story board, which can be found in the Day 3 file.

Generally:

- Person hears ad for BitBeatz on Spotify.
- User tells friends, they all start a project.
- Users lock certain controls and make modifications.
- Someone clicks generate.
- Everyone listens to the song and realizes which parts they want to fix.
- They go back and people iteratively make changes.
- Share/save/publish.

June 3, 2020  
Everyone

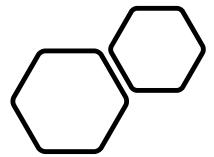
## Debrief of "Monday" - "Wednesday"

- Overall it went well!
- We liked how things came together on day 3, and we now feel on track.
- We disliked working for 3 hours straight without any breaks.
- We found it hard to make decisions and agree on how to move forward.
- We found the art gallery, heat map, and speed critiques could have been shortened, as they were kind of boring and redundant.
- We wish the crazy 8s was slightly longer to get more strong ideas.
- One thing we questioned was having a decider since none of us have spent more time on this compared to each other.

June 3, 2020  
Everyone

## "Wednesday" Summary

- Went through solutions and decided on a few key elements we liked
- Chose a decider (Griffin) to expedite things and produce a template for our solution
- Chose a brand name, BitBeatz, for the product
- Worked out a general-purpose user story
- Decided to focus on UI for prototyping



# "Thursday"

Prototyping

June 5-6, 2020  
Griffin

# Building the prototype in Figma

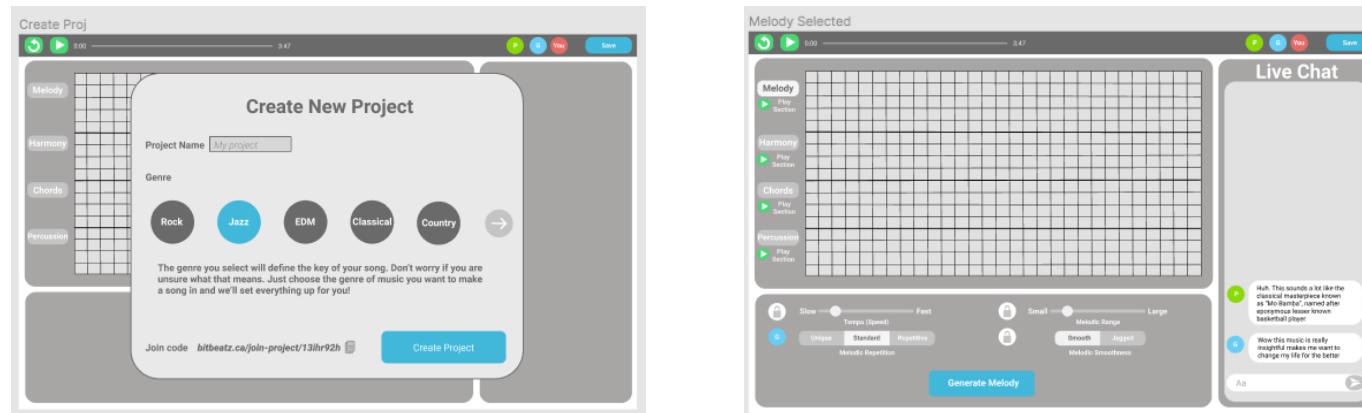
I worked on the following components in Figma:

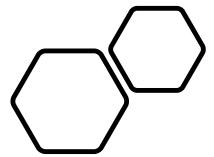
- Create new project screen
- Controls section with locks
- Top bar with play button, etc.
- Chat

I also helped make it a usable prototype by linking up the components.

It was cool learning how to use Figma since I had never used it before. It has proved to be a very useful tool.

<https://www.figma.com/file/W6R2ZgZv04BEALIMVwP6wJ/Prototype-screens?node-id=0%3A1>





# "Friday"

User Testing

# Testing on Figma (User 1: 20M)

June 10, 2020  
Everyone  
Notes by Griffin

## Pre Questions:

- Doesn't know any platforms to create music as an amateur
- Thinks overwriting each other's work could be a problem for working together online
- Doesn't want to see way too many settings (keep closer to 5 settings rather than 50)
- Most enjoyable part? Being able to share music is essential. Wants visual feedback from friends listening to music (smiling, laughing).

## Login:

- standard
- missing forgot password

## Create Project:

- feels accessible with genres
- should be able to get back to create project screen after clicking start

## Main UI:

- Live chat is liked
- Should be able to minimize live chat
- Title above features instead of under
- Some buttons are unspecific, expects more explanation
- length of song is confusing since it looks like a loop, with no where to edit length
- wants to be able to change melody etc from grid as well as bottom
- thinks it might be hard to coordinate the locks over just chat
- should have a share button or something up top to get access to the share link again

## Post Interview:

- enjoy most?: the simple UI, it's not too cluttered, fairly self-explanatory
- fulfilled prior want for simplicity?: would appreciate little helper text for musical terms
- change?: ability to add and remove controls at the bottom
- enjoyed least?: only collaboration is over chat, would appreciate a live audio call. Would like it built into app rather than third party so that music can be played all together. Feels like he's generating music himself, not collaborating.
- features that worked well together?: being able to play individual parts of music individually. Wasn't sure if user can lock down whole melody section, etc.

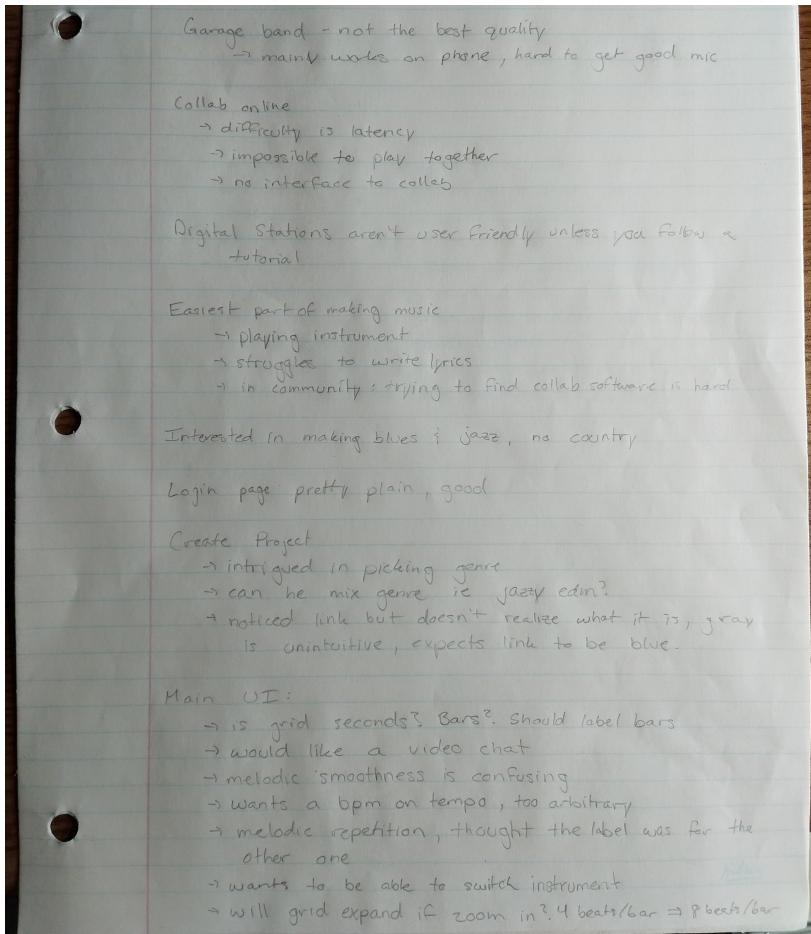
# Testing on Figma (User 1: 20M)

June 10, 2020  
Everyone  
Notes by Michael

- likes rap, soul pop, etc.
- likes music creation
- sees issues w/ live editing of some component
- " " " offers help showing new users what to do, too much complexity
- wants to be able to share
  
- Create project screen feels accessible, make join code accessible after (beside song)
- likes live chat, minimize & though
- \* - likes simple UI at bottom, puts others on top
  - ↳ explain each section more
  - good w/ main play/pause functionality
  - generate → chart not intuitive
  - initial time confusion
- \* - hard to coordinate w/ just chat, video/audio built in (doesn't feel intimate)
- explanation of key musical terms
- likes playing pieces individually

# Testing on Figma (User 2: 19M)

June 10, 2020  
Everyone  
Notes by Griffin



→ questioned if he could click different sections without  
→ green p is controlling the tempo, understands the locking is

Post:  
- interested in computer generating music  
- likes being able to select genre since everything varies so much  
- wants to change instrumentation, between genres  
- wants grid to be zoomable w/ units (change between beats and mins)  
- likes live chat since has never used an audio station where you can collab. Wants video chat, but thinks it could be an issue while people try to listen. Maybe be able to mute each other.

# Testing on Figma (User 2: 19M)

June 10, 2020  
Everyone  
Notes by Michael

- Garage band not great
- most digital audio studios aren't all user friendly
  - many issues being latency
  - difficult to find people to collaborate with blues/jazz fan
  - likes genre choosing, curious about mixing
  - share link not obvious
  - not sure format of graph (beats/bars, seconds, etc)
  - more interactive chart options
  - believe smoothness confusing
  - tempo - un.t.b (beats/minute)
  - diff. for bt config confusing
  - trying to intuitive
  - search instrument / synth
  - Ability to expand grid to see more detail?
  - Editing of generated percussion etc. ?
  - Understands tracking :)
  - likes genre specification
  - wants instrument specification
  - needs better grid - timing, zooming, markings
  - live chart - interesting, text & video path important

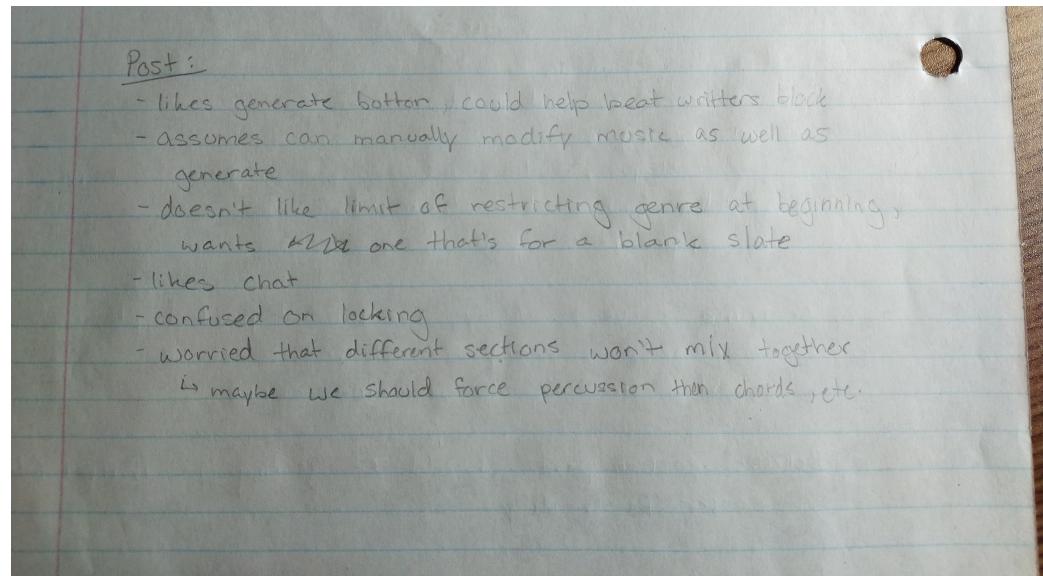
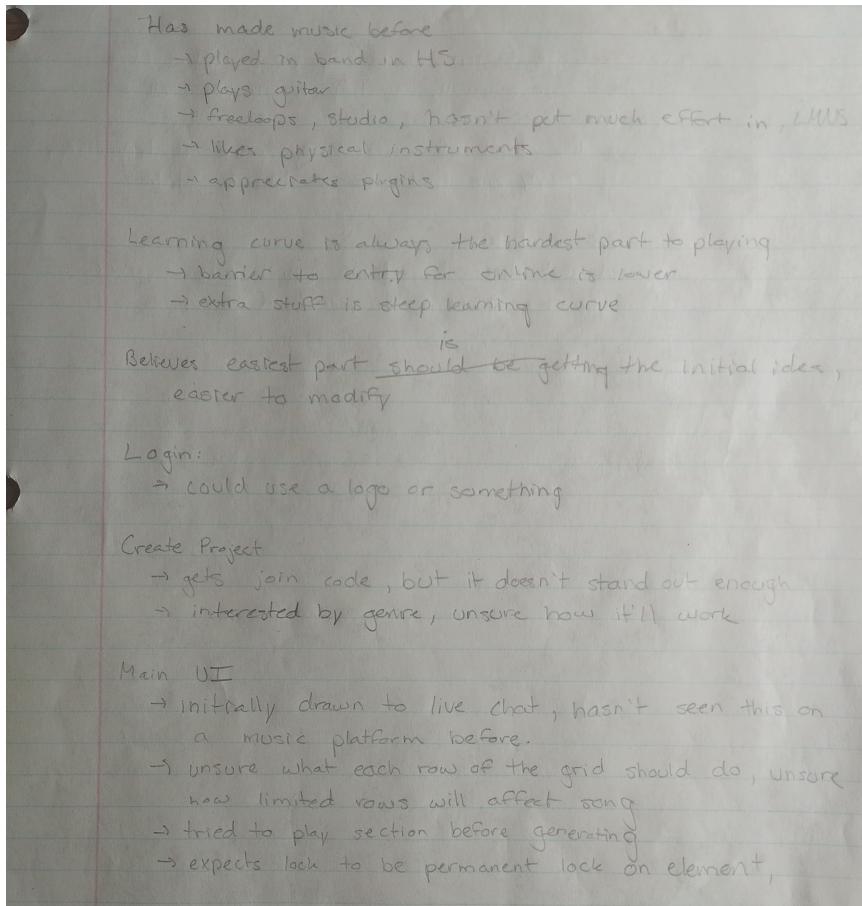
# Testing on Figma (User 2: 19M)

June 10, 2020  
Everyone  
Notes by Peter

- Experienced with music
- Thinks genre selection function is very important
- Would like to see the grid be have more granularity in terms of beats and bars – zoom feature?
- Thinks adding video chat will make it easier to collaborate

# Testing on Figma (User 3: 21M)

June 10, 2020  
Everyone  
Notes by Griffin



# Testing on Figma (User 3: 21M)

June 10, 2020  
Everyone  
Notes by Michael

- Enjoys physical aspect of music making
- Appreciates ability to alter music
- Sees learning curve as biggest barrier to entry
- Easiest part should be getting the ideas out there (brainstorm inspirations)
- understands 'join code', should be easier to see
- genres "interesting"
- likes colour scheme
- live chat is new, drew him to it
- different instruments layer to create percussion
- need more rows
- understands locking well :)
- liked generate idea, so long as manual is possible too
- have blank slate genre option, change pathway?
- lock for multiple people?
- questions of compatibility for components

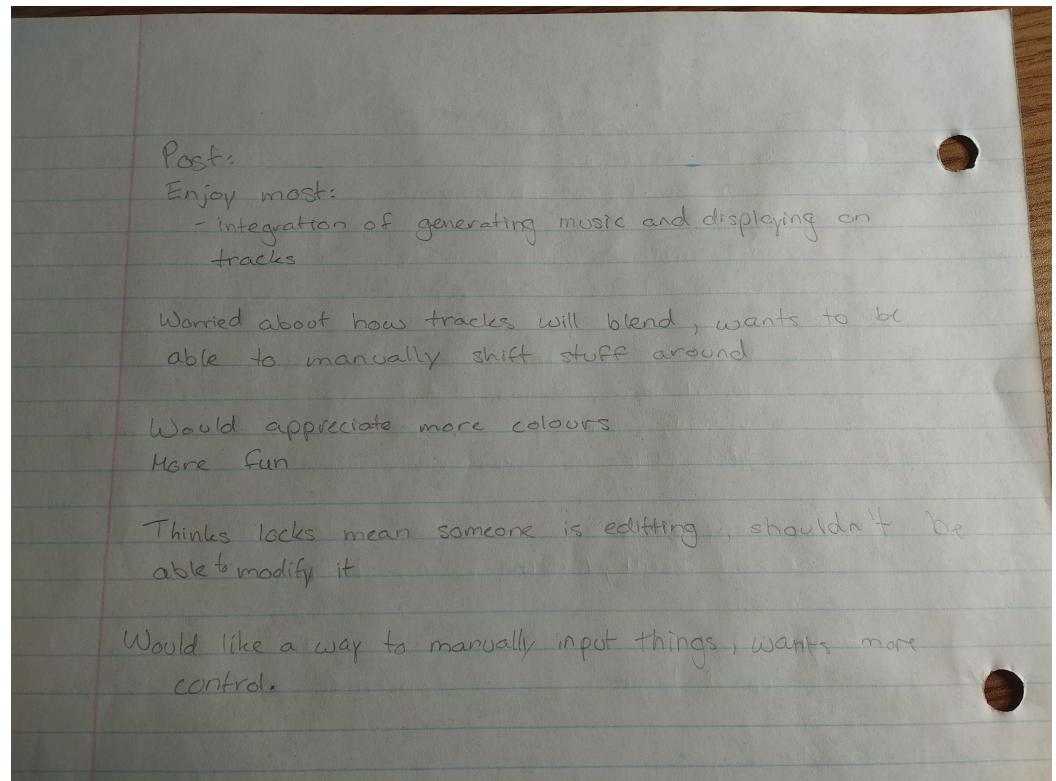
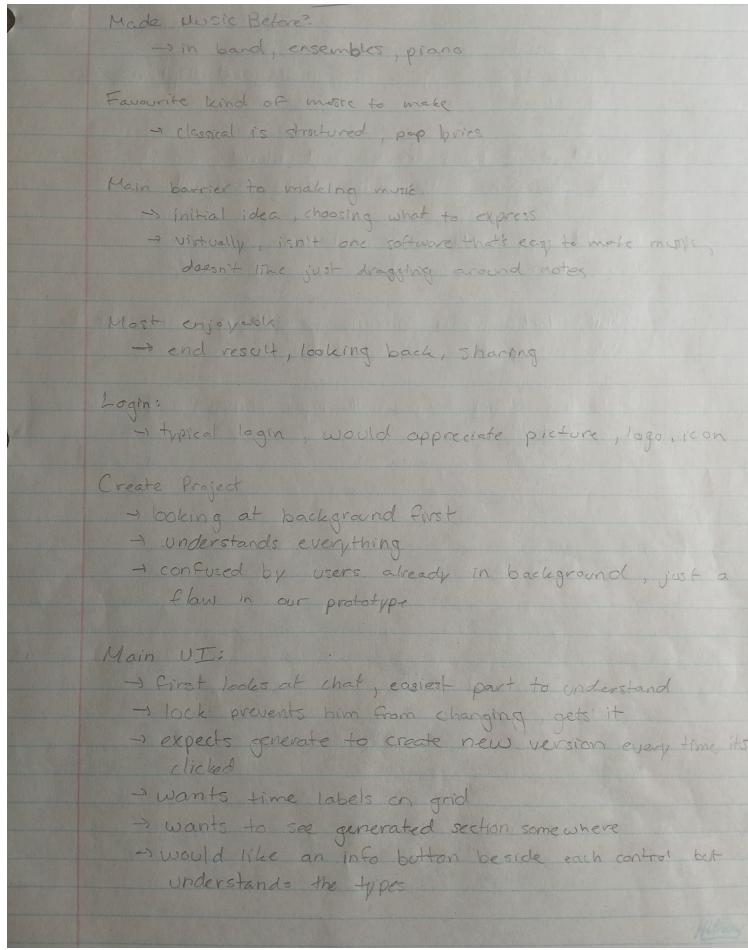
# Testing on Figma (User 3: 21M)

June 10, 2020  
Everyone  
Notes by Peter

- Relatively inexperienced musically
- Thinks that one of the main barriers to creating music is the learning curve or complexity
- Wants to be able to "just get ideas out"
- Wants to be able to do more things manually, rather than have the app do everything automatically
- Wants a blank slate option for the genre, rather than having to choose a specific one

# Testing on Figma (User 4: 20M)

June 10, 2020  
Everyone  
Notes by Griffin



# Testing on Figma (User 4: 20M)

June 10, 2020  
Everyone  
Notes by Michael

- main barrier is idea generation
- add a logo
  - understands locking well :)
  - very enthusiastic, good user
  - regenerates when clicked again
  - info buttons
  - blending can be an issue

# Testing on Figma (User 4: 20M)

June 10, 2020  
Everyone  
Notes by Peter

- Relatively experienced with music
- Largest barrier to creating is coming up with ideas
- Most enjoyable part about creating music is the end result and sharing it
- Wants to see markings on the grid to show the beats or time signature
- Thinks an info button would be useful to explain some of the less intuitive parts of the UI
- The lock system for the UI is not intuitive

# Testing on Figma (User 5: 23M)

June 10, 2020  
Everyone  
Notes by Griffin

- Never ~~played~~ made music before  
↳ has tried playing guitar

- Likes alt, indie. Would be interested in playing this genre

- Main barrier to making music?  
↳ perseverance to learn and create

- Easiest part?  
↳ "when remaking its easier to choose a sample"

Login:  
↳ standard login for music service

Create New Project  
↳ understands page

Main UI  
↳ initially clicked generate  
↳ expects grid to show music  
↳ expects generate to generate everything  
↳ clicked play button for section before clicking generate  
↳ gets play button at the top, with restart  
↳ unsure what harmonic repetition etc. means  
↳ thinks you

Post  
Enjoy most? Really likes live chat, but would like voice chat as well  
Enjoy least? Feels unexperienced  
Feels as though it isn't intuitive enough for amateurs  
Wants a tutorial or hover info to explain stuff  
Play bar at top seems detached from other playing  
Would appreciate more colour in UI

# Testing on Figma (User 5: 23M)

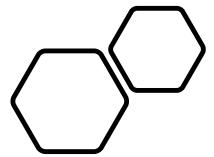
June 10, 2020  
Everyone  
Notes by Michael

- never made music
- doesn't know anything about making music
- barrier to entry: Creativity & time to invest
- join calls okay
- genres okay
- drawn to play more than generate
- generate all at once?
- unsure of what settings mean? → needs clarification
- difficulty finding settings → titles
- liked live chat, wants voice chat :)
- grey bar groups & other components
- more colour :)

# Testing on Figma (User 5: 23M)

June 10, 2020  
Everyone  
Notes by Peter

- Inexperienced with music
- Likes the live chat feature, but thinks voice chat would be better
- Feels like his lack of musical knowledge would hold him back from creating
- Would like a tutorial that explained some of the basic concepts of the application
- Location of the play button is confusing
- Would like more colour in the UI



# “Recap”

What we learned in Sprint

# Problem Space Summary

In this sprint our goal was to design a platform for collaborative musical creation over a network that integrates automatic music generation, and targets beginners or inexperienced musicians. Our main objective for the prototype and user testing was to determine what makes a good UI for musical creation and collaboration, and our prototype reflected this, focusing mainly on UI features that we thought would support these goals. Our testing was focused on determining how intuitive the features were to users, as well as attempting to find common misunderstandings or points of confusion caused by the UI.

June 12, 2020  
Everyone

# User Testing Successes

Two questions stood out in the interview process, providing the most valuable insight as to how product succeeded and failed in the eyes of the test user. The first critical question was "*What did you enjoy the most about the experience?*", as the question tended to draw attention to the best aspects of the prototype, and provided specific insight as to why whatever feature or process produced such results. The second critical question was "*What did you enjoy the least about the experience?*", as the question tended to draw attention to the worst aspects of the prototype, and provided specific insight as to why whatever feature or process produced such poor results. These two questions have provided focal points for our team to work from when we move beyond the sprint.

The most well-liked and well-understood feature of the prototype presented during user testing was the chat feature. This aspect's function was recognized and understood by all users, most of which identified it as an essential tool to achieving the goal of successfully creating collaborative music. For multiple composers to work together, it is essential they have the means to communicate clearly. Moving forward, some users suggested bringing communication a step further via a voice chat feature, although this may present additional challenges such as overwhelming audio cues.

The most valuable test activity conducted was having the users describe their thought process as they modified meters in the genetic algorithm generation section. By directly affecting the interface and having a visual result, it was clear what parts of the section were intuitive and what parts needed to be clarified. For example, all users understood the function of the tempo slider after being able to interact with it, but very few could identify the purpose of the melodic smoothness meter, even when given built in visual cues. This led to the suggestion of adding a tutorial or info boxes to help beginner users better understand what they can and cannot do when modifying the rules of the genetic algorithm.

June 12, 2020  
Everyone

# User Testing Failures

The least valuable question asked, which provided the least insight overall throughout the process was “*What do you believe will be the easiest part of making music?*”. The question seemed to lack a specific focus and yielded varied and vague answers. Overall, it seemed to create more filler than legitimate insight. It may have primed the user for beginning their interactions with the prototype, but there was no way to measure that.

The least-liked and least-understood part of the prototype was the grid representing a visual picture of the created music. While all users were able to identify it as some sort of visual cue, it did not help any of their understandings and even confused users in some cases. This grid was interpreted by test users as a placeholder for soundwaves, MIDI files, individual tracks, and individual notes (the intended interpretation). Having this section as an empty canvas rather than filled in with generated music created unnecessary confusion, and better results could have been achieved had sample pieces been filled as a representation of its purpose.

The least valuable test activity conducted in our user testing was having the user ‘generate algorithms’ and listen to their created pieces. The intent was to have users click the ‘generate algorithm’ button for each of the four components of their created song, and then play the completed song at the end in order to better understand how the different aspects of music they could create would all fit together. However, all users, even those with little to no music experience, were able to grasp and even explain how this part would work before it was tested. In this case the intuitive knowledge of our users was underestimated, making that section of testing redundant.

There were distinct splits and similarities between the successes and failures in testing the musically and non-musically experienced users. All users, regardless of experience, enjoyed the live chat provided in the prototype – this proved a shared success. The non-musically inclined users tended to be more confused when it came to understanding the controls and didn’t ask as many questions about the interface. This may have been because there was a lack of functionality in the basic prototype. The musically inclined users were confused by the interface and tended to ask a lot more questions about the technical functionality as they attempted to reconcile the interface at their disposal with past tools they had used. The difference in frequency and targeting of questions from the user most likely came from the fact that the musically inclined had the knowledge to ask those specific questions, while the non-inclined didn’t have that base to work from. In the future, to better understand those without experience, a stronger suite of test questions to be asked throughout the interview should be developed to better gauge the cognition of users who may not be able to articulate their specific concerns.

June 12, 2020  
Everyone

## *“How can we effectively test this product without bias, and what are our standards for success?”*

The sprint process proved to be an effective means for providing answers to the questions we had about building our product. One of the biggest questions we had for effectively executing the iterative design process going forward dealt with testing; “*how can we effectively test this product without bias, and what are our standards for success?*”. The end goal for the product is to solve the problem but that only guides the product and doesn’t answer how we measure the success of each iteration of the product, or how well solved the problem is with each design. Testing users with all levels of experience proved useful and is something that should be continued with all future testing. A good metric to use for determining overall user experience with the product is the Net Promoter Score, but this “does not necessarily give you information to pinpoint the factors that would improve your actual experience”. [1] A high NPS is a good end goal for the product but the NPS itself is not very informative of the direction future iterations should follow. The interviews made it clear that there are some common frustrations with digital music making tools and other applications that need to be overcome for a successful product. Testing against these elements of the user experience can be an effective way of determining the quality of the design.

[1] "How to Leverage NPS for Greater Retention and Product Stickiness", Qualaroo, 2019. [Online]. Available: <https://blog.qualaroo.com/2019/03/26/nps-a-baseline-for-understanding-the-user-experience/>. [Accessed: 12- Jun- 2020]

# Learning Curve and Testing

One of the greatest frustrations about music making that our interviews brought to light is a common issue with many tools: the learning curve. Users “want to get something done and they will only take the time to learn your product if they think it will help them do so.” [2]. Users expect to be able to quickly get up and running, and don’t want to learn more than necessary, especially when considering Jakob’s Law, “Users spend most of their time on other sites. This means that users prefer your site to work the same way as all the other sites they already know.” [3, chap. 1, “Jakob’s Law”]. One type of testing for this is called a learning experiment, where users perform the same task multiple times to provide quantitative metrics on the rate at which users learn the system [4], and is a technique that could be easily adopted for future interviews. Another testable element that users find frustration within music making specifically is the ideation phase; many users expressed that they enjoy making music but coming up with an idea can be a challenge. To test how helpful the product is with this, we can see how often users keep results generated by the GA, or even whether future states of the music have underlying similarities to any previously generated music if the user is choosing to use the music subconsciously after discarding it. This will also reflect on the overall quality of the music generating algorithm, which we want to test for. There is lots of work to do to create a product that conforms to our criteria for success, and knowing how to test for those criteria to inform future design decisions will be helpful for achieving this.

- [2] B. McKenna, "The learning curve design problem", Medium, 2020. [Online]. Available: <https://uxdesign.cc/the-learning-curve-design-problem-4d4dc2965098>. [Accessed: 12- Jun- 2020]
- [3] J. Yabonski, Laws of UX. O'Reilly Media, Inc, 2020. [Online]. <https://learning.oreilly.com/library/view/laws-of-ux/9781492055303>
- [4] R. Budi, "The Power Law of Learning: Consistency vs. Innovation in User Interfaces", Nielsen Norman Group, 2016. [Online]. Available: <https://www.nngroup.com/articles/power-law-learning/>. [Accessed: 12- Jun- 2020]

June 12, 2020  
Everyone

# *“What are the requirements for successful real-time musical collaboration over a network?”*

Another big question we had before the sprint was *“What are the requirements for successful real-time musical collaboration over a network?”*. During the sprint we approached finding a solution to this problem from the side of the user-interface and user-experience. We did not go into depth on how these features could be implemented and optimized in the backend.

During the sprint, our team realized that we needed to create an intuitive way for users to view where their fellow collaborators are working in real time. We were concerned that without knowing where their group members are working, it would be easy for users to overwrite each other’s work, and the whole process of making music could feel more like an individual project rather than a collaborative project. We contemplated taking a turn-based approach to solve the issue of overwriting each other, but we worried that users would get bored when it wasn’t their turn, and it could turn into a collection of personal modifications rather than actual collaboration. We ended up addressing this issue by mocking up a system that allows users to temporarily select a control they want to modify, which would disallow others from modifying it at the same time. Users would be able to see that someone else is modifying a control when a user icon appears beside the control (see the blue “G” in the figure below). When there is an unlocked icon, the user would be free to modify that control.

This approach ended up confusing users during our user testing, with one user thinking they could lock multiple controls at the same time, and one user thinking the locks would permanently “lock in” a setting for the final song. We think using a lock as the icon could be causing some of the confusion for our users; however, we may need to attempt completely different methods for displaying user location in future iterations. Whatever method we choose, we will want to ensure that our interface provides feedback from the system within 400ms if we want to keep the user’s attention and increase productivity [3: chap. 10, “Doherty Threshold”].

[3] J. Yabonski, Laws of UX. O'Reilly Media, Inc, 2020. [Online].  
<https://learning.oreilly.com/library/view/laws-of-ux/9781492055303>



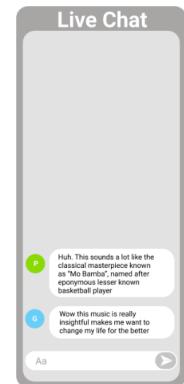
Screenshot of the controls from our prototype

June 12, 2020  
Everyone

# Communication Feedback

In order to implement real-time collaboration over a network, our team also decided that we need methods of communication built into our application. We implemented a live text chat in our prototype, allowing users to communicate back and forth in real time.

Users liked the ability to communicate within the app, some saying that it set our design apart from what they had previously used to create digital music. Although users appreciated the text chat, they felt as though the ability to communicate by voice would improve the user-experience and reduce possible headaches when trying to communicate more complex ideas. A couple users also mentioned that they would even appreciate a video chat built into the application, allowing them to read their friends' facial expressions, since the sharing aspect is their favourite part of making music. Implementing these methods of communication should be a strong focus in future iterations of our design.



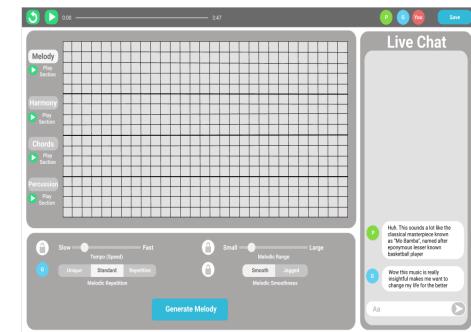
June 12, 2020  
Everyone

# User Interface Reflection

During this sprint, our prototype design and testing were all centred around the UI (User Interface) of our software. One of the most important questions that we wanted to answer was “*What makes an effective musical UI?*”. This question is based on our idea that one of the most important parts of good software is a good UI, as well as the idea that the type of UI that is best for creating music may not be the same as it would be for other tasks.

We decided that some of the most important UI features for creating music are real-time feedback (both visual and audible) of the current state of the music, and intuitive links between this feedback and the controls, as well as the intent of each control being clear. In order to implement real-time feedback in our UI we used a grid divided into 4 sections, one for each of the Melody, Harmony, Chords, and Percussion portions of the project. In the final product each row of squares will represent a note, with the columns representing beats or sub-beats. This was hard to understand for the users during testing, but we believe this was due to the lack of visual or audio representation due to the limitations of the prototype.

Each of the labels for Melody, Harmony, Chords, and Percussion open their corresponding controls panel when clicked. This makes it clear which controls correspond to each section of the project.



June 12, 2020  
Everyone

# Controls Feedback

During user testing we determined that our controls were potentially confusing to users, especially those who were less musically experienced. This led us to the conclusion that we need to make the meaning of each control clearer, potentially by adding info buttons that describe what each control does in understandable terms for users with little musical background. This problem could also go away with a higher fidelity prototype where the correlation between manipulation of the controls and the resulting changes to the music are more obvious. This real-time feedback is important in order to enable further testing of the control functionality. Making sure the intent of the controls, as well as the intuitive connection between control actions and changes in the music are obvious to the user should be heavily emphasized in our next iteration.

June 12, 2020  
Everyone

# *“How much musical knowledge should users require to be able to use our product?”*

Another question we had before the sprint was “*How much musical knowledge should users require to be able to use our product?*”. We discussed possible answers to this question early in the sprint in order to determine how much complexity we would incorporate in our prototype. We decided that our target user for the sprint should be someone with little musical knowledge because theoretically if they can use the application then so could an experienced user. When we conducted our user testing, we had two users with little to no musical knowledge, but we also had three users with more experience in music theory and using digital music interfaces. Every user we interviewed found the program intriguing in different ways, so we believe that it is important to make the program accessible to amateurs but with enough functionality to keep more musically experienced engaged.

June 12, 2020  
Griffin

# Individual Reflection

After completing the sprint, our team has collected enough information to develop some answers to our original questions. One question we had was "How much musical knowledge should users require to be able to use our product?". I think it should be our goal to make a product that is accessible to people with low musical knowledge, without impairing the workflow of a more experienced user. The novice users were able to create music, however they did not feel as though there was enough description within the control titles to fully understand what they were changing. Some of our more experienced users also found some titles confusing. A few users stated that they would appreciate a tooltip for more information on how their changes were affecting the music; however, tooltips should not include essential information for users to complete a task [4]. In the future we should try rewording our control titles and possibly adding a tutorial for beginners.

Another question we had was "What are the requirements for successful real-time musical collaboration over a network?". What I took away from the sprint is that we need an intuitive way of representing where other users are located on the application, as well as a variety of methods to communicate. We attempted to replicate physical cues by placing user icons beside controls, representing which user is changing a part of the music, and restricting others from making changes to the same control at the same time; however, a few users found this locking feature confusing and unintuitive. We also implemented a text chat along the side of our application, which was generally liked; nonetheless, many users that tested our prototype mentioned it can be difficult to communicate bigger ideas in a text chat, and would appreciate a voice chat, or even a video chat as well. These methods of communication will be important to focus on during future iterations of our design.

The last main question we had before our sprint started was "What are our standards for success and how can they be evaluated and tested?". After the sprint I would say our standards for success have become heavily focused upon having a user-interface that is intuitive for users of various skill levels, and allows users to collaborate seamlessly. We will know our product is successful when a musical beginner is able to collaborate with a musical expert and create a song that they both enjoy. We need to ensure that we keep testing users of various skill levels, and it would be good to test diverse users in groups, as collaborators, rather than as individuals.

[4] A. Joyce, "Tooltip Guidelines", Nielsen Norman Group, 2019. [Online]. Available: <https://www.nngroup.com/articles/tooltip-guidelines/>. [Accessed: 11-Jun- 2020].

June 12, 2020  
Everyone

# Team Contract

**Team Name:**

Team 10 - BitBeatz

These are the terms of group conduct and cooperation that we agree on as a team.

**Participation:**

We agree to provide our full attention to group activities and contribute an agreed upon amount of effort representative of the total work required for deliverables.

**Communication:**

We agree to express our honest views to the team and bring to light any issues that may be detrimental to the team as soon as we become aware of them. We will speak plainly about them.

**Meetings:**

We agree to meet at the agreed upon time for the agreed upon timeframe and for longer if absolutely necessary to complete the deliverable or group activity

**Conduct:**

We agree to act with professional conduct as necessary to complete deliverables

**Conflict:**

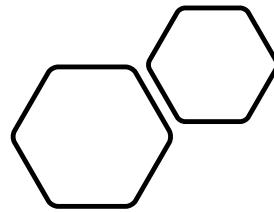
A gentlemen's game of rock paper scissors will be initiated over Microsoft teams. Best two out of three of course.

**Deadlines:**

We agree to work to ensure milestones for deliverables are met to ensure that deadline times are also met

Team Member's Name	Team Member's Signature
Michael Lipski	MW lpsi
Kevin McGuire	Kel
Greg Hoerdt	Greg Hoerdt
Peter Marshall	Peter Marshall
Thomas Neter	Thomas Neter
Griffin Cook	Griffin Cook

# Second Design Iteration



June 18th

Kevin

# GA Meeting

- Greg and I met to discuss developing a game plan for the developing the GA
- Greg developed the initial plan to focus on tailoring the GA for jazz music, outlined the fundamentals of Jazz percussion composition.
- Rough sketches of chromosome composition for the percussion class were exchanged

# Jazz Persuccion Research

Common attributes among standard Jazz drum lines:

Genetic Algorithm Template: Percussion

-Input 8 'on' values into a 5x6 placeholder grid

-green and white squares are fixed; green is always on, white is always off

-between all the orange and red squares, only 2 can be on at a time. 95% of these will be orange

Ride	Green	White	White	Green	White	Fixed 90% of the time	White
Bass	Green	White	White	White	White	Fixed	White
HiHat	White	White	White	White	White	Fixed	White
Snare	Yellow	Red	Yellow	Yellow	Red	Yellow	random, only 2 on in this 2x6 grid - 95% orange, 5% red
Tom	Yellow	Red	Yellow	Yellow	Red	Yellow	random, only 2 on in this 2x6 grid - 95% orange, 5% red

<https://www.youtube.com/watch?v=qz1EOtI-R3E>

<https://www.youtube.com/watch?v=G13NEVAm6-o>

<https://www.youtube.com/watch?v=4tAm3GThdBA>

# Jazz Persuccion Research Continued

Plan for adapting genetic algorithm method to percussoin matrix:

Each space in the grid represents an anelle, while each grid represents a chromosome

GA method Summary

1. 8 input 'ON' signals will be distributed randomly amongst a 5x6 grid
2. This is repeated several times to make a 'generation'
3. The generation is divided into pairs
4. Specific pairs are selected based on their fitness (Number of 'on' values in acceptable positions based on the colour grid explained earlier)
4. Selected pairs perform a crossover (Still working out how to do this with a matrix)
5. Each new grid performs a random mutation (a random 0->1 and a random 1->0)
6. The results become the new generation, and the process repeats until a prescribed number of desirable grids are created
7. This process outputs a repeating improvised jazz drum line

June 18th  
Kevin

# Prototype

Rudimentary GA functionality implemented to completion in python.

- All functionality is implemented, and population stays steady between generations.
- Modifying fitness function weightings to observe effects
- Changing function names to be less comedic, more professional

June 18th  
Kevin

# Prototype

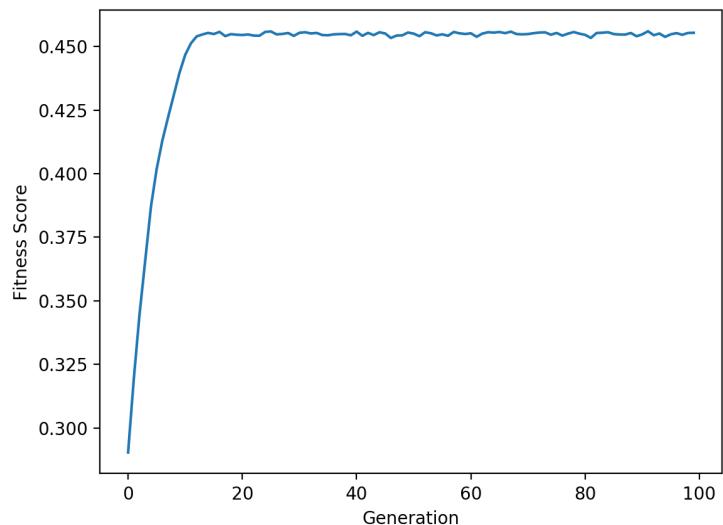
Rudimentary GA functionality continuing to be implemented in python.

- Basic Percussion Class modified for extended array capabilities
- Combination function fixed.
- Generation numbers are unstable, decaying, issue is within the selection function I believe

June 18th  
Kevin

# Prototype

Ga is training to quickly, attempting to play around with the mutation setting to alleviate, no luck so far.



Note distinct plateau

June 19, 2020  
Everyone

# Prototype Plan (slide 1/5)

**SOC:** Creating music collaboratively online is challenging with current tools due to poor networking functionality, and users need to spend a significant amount of time getting accustomed to using these tools. Current tools require that users must also have a background in music to create music and that music creation be done manually. This creates a barrier to entry for amateur musicians and requires time investment for all users when creating a piece of music.

**SIS:** The final product will positively impact the mental health of our user base, allowing them the basic human connection only fostered through the creation of music. This primal need will be fulfilled and sated through our platform.

## Design Options:

### Improvements from LFP for both versions:

- Implement voice chat alongside text chat.
- Create better labels for controls.
- Add a button to be able to return to project setup screen with share/edit link.
- Create a share/listen button that exports the file for others.

#### 1. Basic Version

- Designed with beginners in mind, meant to have a low learning curve, while sacrificing some creative freedom.
- GA generates all 4 parts (percussion, chords, melody, harmony) together.
- Settings and inputs are basic, only using sliders, dials, and buttons.
- Simplistic UI for visualization of generated music (Wave).

#### 2. Pro Version

- Designed for more advanced musicians, with a higher learning curve in order to support customization and creative freedom.
- GA generates all 4 parts (percussion, chords, melody, harmony) separately. We want to reorder the position of these parts in our UI so that percussion is the top element followed by chords, then melody, then harmony.
- Settings and inputs are more advanced/detailed, allow for midi or mp3 inputs.
- UI for music visualization is more granular using the grid design.

June 19, 2020  
Everyone

# Prototype Plan (slide 2/5)

## Specifications:

1. Learning curve – The time to get used to using the product efficiently should be as small as possible. One type of testing for this is called a learning experiment, where users perform the same task multiple times to provide quantitative metrics on the rate at which users learn the system
2. Accessibility – Application should follow relevant sections of the WCAG (Web Content Accessibility Guidelines)
3. Aesthetics will be required as no one will want to use a tool that is displeasing visually. The application must look nice.
4. Low Cost of upkeep – The cost of upkeep for the system should be minimal
5. Latency, system feedback time – The system should have as low of a latency as possible to mitigate user frustration with using the product and facilitate efficiency of use
6. The GA must create sounds that fit together as music, rather than a random collection of sounds.
7. OWASP (security standards) – the system must be compliant with OWASP 2 to ensure the security of the users
8. Must allow for users to communicate within the application.
9. Must not output a volume above 70dBA for safety reasons [<https://www.noisyplanet.nidcd.nih.gov/parents/too-loud-too-long#:~:text=Sounds%20at%20or%20below%2070,greater%20risk%20for%20hearing%20loss>].
10. Must meet MIDI standards.

# Prototype Plan (slide 3/5)

Visual Representation:



June 19, 2020  
Everyone

# Prototype Plan (slide 4/5)

## **Team Member Roles:**

(Team members may shift around a bit to support each other in areas that require more work, but these are the assigned roles)

Michael – GUI development, integration (of GUI, GA, Network), PMO

Kevin - GA Development

Griffin- GUI Design/Development

Peter – Networking

Thomas – GUI development

Greg – GA Engineer, Product Manager

# Prototype Plan (slide 5/5)

## Feedback from Prof. Borland

- Good start. More detail would have let me give you better feedback.
- Your SIS could be more specific. It's a bit high-level...what are you actually doing?
- I was hoping to see a stronger visual representation of the plan, the paths and goals, that will see you compete the project.
- Some of your specs are needed, but not that important. I would definitely recommend using an existing piece of software that receives MIDI messages and turns them into sounds instead of building your own.
- I don't require a fully functioning, fully connected end to end experience. MFP's can be built in chunks that demonstrate functionality separately.
- GA's to control all aspects of music making will be challenging - moving from percussion tracks to melodies and harmony will require exponential increases in complexity and GA rules.
- Are there other more important aspects of the solution it would be good to measure and compare?
- Is using voice and text chat for collaboration a missed opportunity? What is the language/syntax of real time musical collaboration, and how can those things be represented/communicated digitally? Is text or voice chat the only/best way?

## Changes to the plan:

- Focus on percussion rather than trying to incorporate melodies, harmonies, and chords too.
- Shift the second prototype to be the same as the first one just without locks so that we can test if they are a good method of communication.
- Add Peter as another UI developer.

June 20th  
Kevin

# GA Selection Research

Thought that I might eventually need to scale up the complexity of the selection algorithm.

Spent time researching other ways to select pairs

This resource proved interesting:

<https://www.obitko.com/tutorials/genetic-algorithms/selection.php>

May implement a way to allow less fit members to reproduce

# AMA Questions

We came up with questions for the AMA with Prof. Borland's musician friends. The questions were:

- Are there certain physical cues you rely on for communicating while playing music with others? For example, watching another musician's hands playing their instrument, nodding at each other, pointing for timing.
- Do you normally collaborate on creating the percussion, chords, melody and harmonies in the same creative session, or is it more so one person will create the percussion, then someone else will create chords that go well with the percussion and so on? Or have you found that this varies on the song and group?
- Generally when a group of musicians are writing a song, what kind of stages do they go through, and what kind of standard roles pop up (how does leadership work etc.)? Does someone typically make the large stylistic decisions, or is a more democratic approach present?

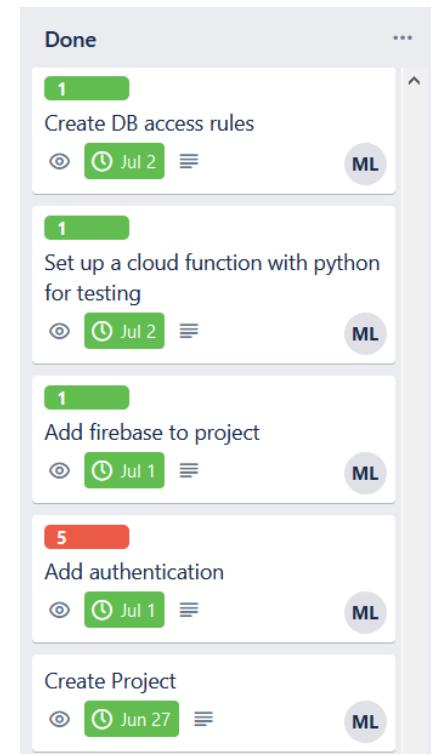
June 27  
Michael

# Prototype work - Michael

I created a Trello board for the team to track tickets with if they'd like. Trello is a very helpful organizational tool for keeping track of things to do and is often used in Agile sprint methodology. I begin by creating tickets where a description is provided and then I provide estimates designated by the labels 1, 2, 3, 5. Tickets are shown in user-created columns such as the one on the right. These are put in the to-do column, then I move them to "in progress" when I am working on them, then finally to "done" when the corresponding commit for the work has been pushed to GitHub and verified. When tickets are moved to done I set the due date on them as a simple means of tracking the times.

This was how I tracked my work personally, and git commits can be used to verify that the work was completed. This was a useful exercise in project management for me personally, and is something I will likely continue doing in 362 and capstone.

All prototype slides show the tickets and descriptions of what I worked on in roughly chronological order.



June 27 - July 2  
Michael

# Prototype

Project architecture work (setting up the project and infrastructure so the front-end and back-end teams could get their work done)

The screenshot shows a Trello board with five cards:

- Create Project**:
  - in list Done
  - MEMBERS: ML
  - DUE DATE: Jun 27 at 2:01 PM (COMPLETE)
  - Description: Create a project for github, firebase
- Add firebase to project**:
  - in list Done
  - MEMBERS: ML
  - LABELS: 1
  - DUE DATE: Jul 1 at 2:01 PM (COMPLETE)
  - Description: Configure front-end for firebase
- Add authentication**:
  - in list Done
  - MEMBERS: ML
  - LABELS: 5
  - DUE DATE: Jul 1 at 2:01 PM (COMPLETE)
  - Description: Create login, hook up with firebase auth
- Create DB access rules**:
  - in list Done
  - MEMBERS: ML
  - LABELS: 1
  - DUE DATE: Jul 2 at 2:01 PM (COMPLETE)
  - Description:
    - We need to ensure only logged in users with access to the project have read/write access to that project
    - We need to disallow read/write to all other projects and users

\* See June 27 slide “Prototype Work – Michael” for ticket management methodology using Trello

# Gurpreet Chana AMA Answers

July 1, 2020  
Notes taken by Griffin

Gurpreet Chana Q+A:

Cues:

- ↳ changes w/ musicians but being in room together helps a lot
- ↳ could just see beginning or end of a cue
- ↳ depends on personality, tone, stuttering help communication
- ↳ can play w/ people for first time on stage, but culture of players is a thing (ie waving guitar)
- ↳ sometimes you don't hear each other, or in recording studio, hunch when quieter, stand bigger when louder.
- ↳ sometimes explicit stuff like calling out chords

General Thoughts:

- ↳ How we converse through music is important
- ↳ Learning from here will help other industries

Tools Rx:

- ↳ Mac Computer, standardised baseline
- ↳ Audio interfaces, 2-channel, multi-channel
- ↳ Zoom, one can share audio from computer as source
- ↳ Splice → live sessions
- ↳ WeTransfer, Dropbox

Electronic Music Interpretation:

- ↳ composing, writing, performing, publishing electronically

Sheet Music? Left-right? Up-down?

- ↳ Doesn't use sheet music, can use syllables, left-right, up-down
- ↳ Mostly memorized
- ↳ Fixed MIDI, animation of MIDI clip

Instruments often missing?

- ↳ Everyone wants a good drummer & vocalist
- ↳ Depends on part of process you're on
- ↳ Simple drum beat would be awesome
- ↳ Amplify, Blockes good for quick creations
- ↳ Flute is missed a lot
- ↳ bowed string instruments
- ↳ singers and harmonies
- ↳ In the end depends on person.
- ↳ World is a drum sometimes

OSC (open sound control)

Could convert to MIDI.

Live labs, testing head bobbing?

- ↳ McMaster Lab

## Most important takeaways:

- Our question about cues got an answer: body language and vocal tone are key aspects when Gurpreet is collaborating. Even just seeing the beginning or end of a cue can be helpful, and is often what occurs. This suggests that it would be beneficial to use a video chat like some users recommended during our sprint.
- MIDI clips act as a good visualization for the music.
- Interested in a variety of instruments for different situations.

July 2  
Michael

# Prototype

## Cloud function work

### Set up a cloud function with python for testing

in list Done

MEMBERS

LABELS

DUE DATE

ML



1



+



Jul 2 at 2:01 PM

COMPLETE



#### Description

refer to [https://cloud.google.com/functions/docs/calling/cloud-firestore#functions\\_firebase\\_firestore-python](https://cloud.google.com/functions/docs/calling/cloud-firestore#functions_firebase_firestore-python)  
<https://cloud.google.com/functions/docs/calling/http>

### 'Create project' back-end

in list Done

MEMBERS

LABELS

DUE DATE

ML



2



Jul 2 at 2:01 PM

COMPLETE



#### Description

When a user creates a new project, we need to:  
create a new project with a unique id in the DB and initialize it with some values,  
create an invite code (might be same as UID),  
couple the project and user  
save the name, and genre to the project in the DB

### Validate user requests for cloud functions

in list Done

MEMBERS

LABELS

DUE DATE

ML



1



Jul 2 at 2:01 PM

COMPLETE



#### Description

need to check that the token with the request is valid for our firebase app,  
and that they have permissions for that project

\* See June 27 slide "Prototype Work – Michael" for ticket management methodology using Trello

July 2, 2020  
Peter

# Prototype – Setup CI/CD

- Setup automated build + test on each git commit using CircleCI
- Set up Heroku to automatically deploy the latest version of the master branch to a live website
- Purchased the bitbeatz.ca domain and configured it to point to the Heroku webserver to make the application easily accessible for testing

Remove failing create-react-app test in App.test.js

peterdmarshall committed on Jul 2 ✓

Fix circleci config ...

peterdmarshall committed on Jul 2 ✘

Delete yarn.lock, add circleci config

peterdmarshall committed on Jul 2 ✘

HEROKU

Personal > bitbeatz

GitHub Bitbeatz/bitbeatz master

peterdmarshall99@gmail.com: Deployed 4c373f66  
Jul 2 at 1:21 AM · v3 · [Roll back to here](#)

peterdmarshall99@gmail.com: Build succeeded  
Jul 2 at 1:18 AM · [View build log](#)

July 2-3, 2020  
Griffin

# Prototype - Initial React UI work

July 2<sup>nd</sup>:

- Created the project setup template in React
- Created Join Codes based on project ID's from the database on Firebase
- Built empty containers for the Project page, defining the layout

July 3<sup>rd</sup>:

- Fixed some styling with Material-UI to make dynamic
- Created controls component for inputs using Slider components
- Implemented the UI for the locks. They are clickable like radio buttons. Currently only local.
- Created a top bar for the Project page. Holds play button, name of project and share code.

Learned a lot to do with React since I've barely used it before, also read up on the Material-UI documentation to learn how to implement it properly.

Merge branch 'master' of github.com:Bitbeatz/bitbeatz	2ce3bda
griffincook committed on Jul 2 ✓	
container shod	77d1712
griffincook committed on Jul 2	
Merge branch 'master' of github.com:Bitbeatz/bitbeatz	002ea125
griffincook committed on Jul 2 ✓	
join codes	ba7c2a7
griffincook committed on Jul 2	
Merge branch 'master' of github.com:Bitbeatz/bitbeatz	37d83a6
griffincook committed on Jul 2 ✓	
nested components	8d61a17
griffincook committed on Jul 2	
Merge branch 'master' of github.com:Bitbeatz/bitbeatz	37fec2f
griffincook committed on Jul 2 ✓	
done template	b4f2b3e
griffincook committed on Jul 2	
Merge branch 'master' of github.com:Bitbeatz/bitbeatz	a50a4ee
griffincook committed on Jul 2	
ProjectSetup.js	1992a69
griffincook committed on Jul 2	

top project bar	2f184ed
griffincook committed on Jul 3 ✓	
locks ui	d237bc7
griffincook committed on Jul 3	
Merge remote-tracking branch 'origin/master'	dd2b2c9
griffincook committed on Jul 3 ✓	
added controls component	949239b
griffincook committed on Jul 3	
Merge branch 'master' of github.com:Bitbeatz/bitbeatz	cc81c46
griffincook committed on Jul 3	
styling on home and project	59ec78e
griffincook committed on Jul 3	

July 3rd  
Kevin

# Fitness function research

Getting ideas for how to create weights and targets for the fitness function

This resource proved interesting:

<https://nature.berkeley.edu/getz/dominance/GA/gads/gadsdemos/html/gafitness.html>

July 3-5  
Michael

# Prototype

Front-end / DB work (At this point with most of the architectural work done I transitioned to helping with the UI)

**Projects navigation front-end**

in list Done ⓘ

MEMBERS	LABELS	DUE DATE
ML +	2 +	✓ Jul 4 at 2:01 PM COMPLETE ▾

**Description** Edit

We need some way to navigate between the user's projects, home can change later

idea: list them in a side bar, have create new project at the top of the list

**Create chat**

in list Done ⓘ

MEMBERS	LABELS	DUE DATE
ML +	2 +	✓ Jul 4 at 2:01 PM COMPLETE ▾

**Description** Edit

Need a chat on each project  
Just need an array of messages in firestore that FE can subscribe to for updates  
should have user and time stamps attached to each message

**Create controls without locks**

in list Done ⓘ

MEMBERS	LABELS	DUE DATE
ML +	1 +	✓ Jul 5 at 2:01 PM COMPLETE ▾

**Description** Edit

Need a version of the controls component without locks to compare with the locks version.  
Can probably just save as a project variable (hasLocks: true/false)

July 4, 2020  
Griffin

# Prototype – Further UI development including connecting to the DB

- Fixed the general spacing on everything.
- Built placeholders for the video chat feature, including arrows to navigate between the group's cams.
- Fixed global styling to make more consistent between components.
- Connected the Controls and Locks components to the DB. Can update the DB and get updates live.
- Wrote logic to map where other users are located and display an avatar with their initial in place of the lock for that control. Initially created the JSON object with the location as the key and the user as the value but that was buggy, so changed it to have the user as the key and the location as the value. This was better for users not overwriting each other, breaking the state.

Learned how to use Firebase and incorporate its API.

controls hooked up to db griffincook committed on Jul 4	e6edb2e
fixed top bar styling griffincook committed on Jul 4 ✓	2abf351
Merge branch 'master' of github.com:Bitbeatz/bitbeatz griffincook committed on Jul 4 ✓	354cd76
theme fixes griffincook committed on Jul 4	d2b06c5
Merge branch 'master' of github.com:Bitbeatz/bitbeatz griffincook committed on Jul 4	13fbcc3
webcam	fb466ec
styling on project file griffincook committed on Jul 4 ✓	538e54c
fixed padding surrounding all content griffincook committed on Jul 4 ✓	ba7a7ab
fixed showing locked controls when other user makes changes griffincook committed on Jul 4 ✓	be4e173
restructured locations to use user as key, works on unload griffincook committed on Jul 4 ✓	5f42f48
use avatars for locks griffincook committed on Jul 4 ✓	872fd39
lock functionality to db griffincook committed on Jul 4 ✓	8727c97

July 4, 2020  
Peter

# Prototype – UI Development

- Built a 2-D array of buttons to serve as the grid where users can select which instruments play on each beat
- Connected the state of the buttons in the grid to the application state, so that the grid is linked to the database
- Each button state toggles between 1 and 0 every time it is pressed
- Modified layout on project screen to fit grid

Make notegrid layout and labels

 petermarshall committed on Jul 4

Add NoteGrid component

 petermarshall committed on Jul 4

# Prototype – Last UI touches for iteration

- Added the generate button below the controls. Not currently connected to anything.
- Added default jazz grid to the front end so that the grid loads with a preset jazz beat. This was designed in the backend, hadn't made it to the front end yet.
- Set the default loop length to 2 beats (6 columns) in order to reflect the way things were implemented on the backend.
- Connected the grid size to the loop length control. Now able to dynamically make the grid shrink or grow by using the slider. Values are 2,4,8, which maps to 6, 12, and 24 columns in the grid.
- Hide wrenches when the no-locks toggle is switched. This created a second version of the prototype to see if users actually prefer no locks at all.

Merge branch 'master' of github.com:Bitbeatz/bitbeatz	a687b48
grid size	4ab2994
Merge branch 'master' of github.com:Bitbeatz/bitbeatz	649ec31
changedComponentDidUpdate	cc30587
hide wrenches for noLocks	32d6c3d
Merge remote-tracking branch 'origin/master'	4ab725d
dynamic grid size	751392d
Merge remote-tracking branch 'origin/master'	35ddff64
jazz grid, generate button, default loop length at 2	6966d11

# Testing on BitBeatz.ca (User 1: 21M)

July 5, 2020  
Everyone  
Notes by Griffin

Login / Sign Up ✓  
Create Project , thought he might need invite code  
  
Doesn't know term "ride"  
Doesn't understand grid  
Took a few seconds to understand locks  
Doesn't understand random variation  
  
Would like explanations w/ more info hover  
Demo sound for instruments  
Changing tempo could play a metronome  
  
Add time interval for grid → should be beats + bars

Has ownership to music  
Prefers locks  
Locks up on grid too  
Wrench is a good symbol  
  
Webcam toggle is good  
show self in corner of webcam  
  
Likes colours , could make sliders same colour as grid,  
make pink and red  
  
Likes side bar

# Testing on BitBeatz.ca (User 1: 21M)

July 5, 2020  
Everyone  
Notes by Michael

What does ride do?  
What is the grid - what does this mean?  
Took a little while to understand looks,  
but got there  
Thinks he understands everything  
Lives interval? columns  
Amused by his creation - feels like it was all him  
Felt the looks were needed :)  
Wants them in grid so  
webcam controls okay  
maybe small self cam view  
Talk w/out video  
red bars instead of blue, change cliched wrench icon  
colours

# Testing on BitBeatz.ca (User 2: 18F)

July 5, 2020  
Everyone  
Notes by Griffin

- Create New Proj ✓
- Doesn't know where to start
- Doesn't know what grid does
- Thinks grid might amplify whole song
- No idea about random variation
- Labels
- Webcam ✓
  - Doesn't intuitively get arrows on webcam
  - Likes chat
  - Thinks listening to beat and video call at same time might hard
  - Likes simplicity
  - Likes version without wrenches better, extra step
  - Sharing end file wouldn't be essential

# Testing on BitBeatz.ca (User 2: 18F)

July 5, 2020  
Everyone  
Notes by Kevin

- unclear on what to do first
- jazz template, favors rock
- unclear with what to start with
- not sure what clicking those will do (buttons)
- unsure as to what the amplification should be
- tempo controls could use more explanation
  - the things are not moving initially (might want to automatically unlock)
  - questions about the switch
  - bpm, maybe? (question about tempo)
  - no understanding for GAs
  - glitching
  - red squares need more explanation
    - time explanations
  - webcam use unclear, with the arrows, but unsure what you'd be switching between, video and audio
  - color scheme is good, more color below
  - spell out what they want you do, animation guiding
  - more process focused, as opposed to the end goal

# Testing on BitBeatz.ca (User 2: 18F)

July 5, 2020  
Everyone  
Notes by Michael

very confused  
just looking fairly well  
create project fine  
application started glitching :(  
project navigation pretty clear  
music creation mildly empowering  
webcam meaning not immediately clear  
errors not clear, audio to video?  
likes the chat -> listening to best  
=> don't want peers library

# Testing on BitBeatz.ca (User 3: 20M)

July 5, 2020  
Everyone  
Notes by Michael

initial login/signup issue (expected)  
tooltip not great  
copy button for share code  
See share code first, then grid  
→ first clicked around grid

Chat is nice

Mihaly didn't like looks but then understood in context

(F) button means cancel changes → change  
play button could be placed much better

expand sidebar by default

expected app name along the top

Little for the sounds  
red a bit aggressive  
need to introduce the 'GA' concept  
Better video presentation  
tutorial pop-up  
Share your project (just share output)

# Testing on BitBeatz.ca (User 3: 20M)

Likes blue  
Red is aggressive

Clean home page  
Tooltip is hard to read for More Info  
"or" is unclear

Wants copy button  
Share through fb, email etc. buttons

Doesn't understand drum types, wants to be explained

Red (X), assumes discard changes

Doesn't initially understand need for locks, learns after partner modifies w/

Play button is too small, weird place, wants it at the bottom

Side bar start open

Expects App name up top, thought proj name

"Random Variation" is confusing, more info would be nice  
Or popup on first enter

Expects Play to keep playing on a loop

Export to mp3 or another type of share code

Share link rather than code would be nice

Hover over chat icons should give full name

Side arrows flip between people, but would rather see everyone's face at once, see own cam in the bottom

Logout is weird placement

July 5, 2020  
Everyone  
Notes by Griffin

July 5-6  
Michael

# Prototype

Bug fixes based on July 5 testing session

**☒ Redirect to project page after joining**  
in list Done

MEMBERS    LABELS    DUE DATE  
 + + Jul 6 at 2:01 PM

**Description**   
When users join a project they should be redirected to the project

**☒ Unsubscribe from DB listener on project unload**  
in list Done

MEMBERS    LABELS    DUE DATE  
 + + Jul 6 at 2:01 PM

**Description**   
When users go from one project to another the app gets confused when someone edits the first project  
When user navigates away from a project we should unsubscribe from the db listener for that project

**☒ Always store grid as 24 beats in DB**  
in list Done

MEMBERS    LABELS    DUE DATE  
 + + Jul 6 at 2:01 PM

**Description**   
When a user goes from 24 beats to 6 beats back to 24, their selections from when they had 24 beats should be carried over -> changes on 6 beats, 12 beats should be UI only, shouldn't effect actual grid

\* See June 27 slide “Prototype Work – Michael” for ticket management methodology using Trello

# Grid Confusion DLF (slide 1/2)

## Insight:

The grid section is not intuitive to musically novice users. It is intended to be a step sequencer used to input and display beats for all of the percussive instruments.

## Evidence:

- Interviews were conducted with three musically novice users, each lasting approximately thirty minutes. They were all asked to describe how they interpreted the functionality of the grid, and to attempt clicking the grid while talking through their thought processes. Notes were taken by our group throughout the whole interview processes.
- One user didn't understand the point of the grid section at all. They guessed that it was a complicated input to control the amplitude of the entire track.
- Two users were able to intuitively understand the matrix representation of beats within the grid because they had seen similar interfaces before including on the Sonic Charge Mircotonic 3 [1]; however, they were confused by the measurement associated with each square in the grid. They questioned whether each square represented a second, a beat, or something else.
- No users understood the meanings of every single instrument. Terms like "bass" and "snare" were generally recognized, but users were confused by the terms: "ride", "hi-hat", and "low tom". One user realized that these labels were all instruments and was just unsure of the sounds they made, but the other two users were unsure if they were even all instruments.

RISE	■	■	■	■	■
BASS	■	■	■	■	
HI-HAT			■		
SNARE	■				
LOW TOM	■				

[1] "Microtonic," Sonic Charge, 2020. [Online]. Available: <https://soniccharge.com/microtonic>. [Accessed: 06-Jul-2020].

July 6, 2020  
Griffin

# Grid Confusion DLF (slide 2/2)

## **Recommended Action:**

### **Greg (Musical Expert):**

Design intuitive labels for the beats on the grid. This includes the specific values and units.

### **Web Dev Team:**

- Decide on the most user-friendly location for the newly designed beat labels and add them to the web application.
- Design and implement a dynamic visual indication of which beat is playing while the user listens to their created beat. Other step sequencers like the one found in the Sonic Charge Microtonic 3 [1] change the colour of a beat when it is playing. This approach could be a good starting point for our next iteration.
- In order to clarify the instrument names, we should play a single note of an instrument when its corresponding name is clicked. We should also include icons of each instrument beside their names. Nielsen Norman Group notes that icons are fast to recognize at a glance, but they should still have visible labels at all times [2].

[1] "Microtonic," Sonic Charge, 2020. [Online]. Available: <https://soniccharge.com/microtonic>. [Accessed: 06-Jul-2020].

[2] A. Harley, "Icon Usability," Nielsen Norman Group, 2014. [Online]. Available: <https://www.nngroup.com/articles/icon-usability/>. [Accessed: 06-Jul-2020].

# Redesign Video Call DLF (slide 1/2)

## Insight:

Users appreciated the ability to communicate with their collaborators over a video call, but the current layout is confusing, and it hinders the ability to read visual cues.



## Evidence:

- We indirectly asked the musician Gurpreet Chana (through a Q+A) which visual cues he relies on when collaboratively playing music. He stated that even small things like the raising of an eyebrow or the tilting of a head can be used to communicate while playing music [3]. This reinforced our need to include video calls within our application.
- Interviews were conducted with three users, in which the video call portion of our application was a focal point for a few minutes each. They were all asked what they thought the arrows on the side of the video call window were meant to do. They were then told that the arrows were meant to toggle between collaborators' webcams and asked if they liked that design. At the end, they were asked if they would be more likely to use the video call or the text chat and why they thought that was the case.
- One user thought the arrows on the sides of the webcam window would toggle between video and audio chat. When told this was not the case, they mentioned having a voice chat without video could be useful.
- Two users mentioned that they would appreciate being able to see themselves in the corner of the webcam window, similar to many popular video call platforms including Skype [4].
- One user stated that having to flip through the different users defeats the whole reason they would want video calls in the application. They mentioned that they want to be able to read all of their collaborators' facial expressions, especially when listening to the music. They did not care if the windows were smaller to fit everyone (Appendix A shows the current size of the video call window relative to the entire application).
- Two users said they would likely use the video call feature when collaborating. One user said they would likely stick to the text chat for communication since they were worried that it could be hard to listen to the music while people are talking.

[3] M. Borland, "Gurpreet Chana Q+A," Jul-2020.

[4] "How do I switch views during a Skype video call?" Microsoft, 2020. [Online]. Available: <https://support.skype.com/en/faq/FA34897/how-do-i-switch-views-during-a-skype-video-call>. [Accessed: 06-Jul-2020].

# Redesign Video Call DLF (slide 2/2)

## Recommended Action:

### Web Dev Team:

- Design and implement a grid view for multiple webcams, including the user's own webcam. The image to the right is an example of this kind of implementation on Skype [4].
- Add the ability to mute collaborators and one's self in the video call. This will make it easier to listen to music from the application without the interference of collaborators' voices.
- Implement and user test automatically lowering or muting the call volume when listening to the music.
- Add the ability to turn off the webcam while staying in a voice chat.



[4] "How do I switch views during a Skype video call?" Microsoft, 2020. [Online]. Available: <https://support.skype.com/en/faq/FA34897/how-do-i-switch-views-during-a-skype-video-call>. [Accessed: 06-Jul-2020].

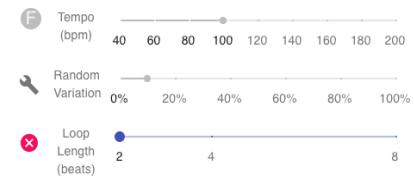
# Disabled Controls are Important but Unintuitive DLF (slide 1/2)

## Insight:

A majority of our users appreciated the ability to disable other collaborators' controls while they made adjustments, but no one immediately found it intuitive. The disabling of controls also added technological stability.

## Evidence:

- Interviews were conducted with three users, in which they were each asked to use two different versions of our application. One version included the ability to individually lock the tempo, random variation, and loop length controls by clicking a wrench icon, causing the selected control to become disabled on their collaborators' screens while they made changes. An avatar with the user's first initial was displayed to collaborators beside the disabled control. The other prototype removed this feature and allowed collaborators to make changes to any controls at the same time.
- Two users really appreciated having the locking controls. One of those users did not initially think it was needed until one of our group members joined their project in the prototype without the locking/disabling and modified the controls at the same time as the user, making the values jump around.
- One user voiced their dislike of the disabled controls since they were working alone.
- Every user attempted to move the disabled controls multiple times before realizing they had to click the wrench to unlock it.
- One user assumed that the pink "x" icon for unlocking the control would actually discard their changes.
- Two users would like to have the locking/disabling feature for the instruments in the grid.
- Our prototype without the locking/disabling feature broke during one test in which a user was collaborating with one of our group members. The control values kept jumping back and forth, causing approximately 8,500 writes and 25,200 reads to and from our database within 10 minutes. This accounted for 81% of our daily reads and writes.



# Redesign Video Call DLF (slide 2/2)

## **Recommended Action:**

### **Web Dev Team:**

- Dynamically remove all of the icons to do with disabling controls, and the disabled attribute from the controls, when there is only one user on the page. Dynamically add them back when a collaborator joins the project.
- Add the control locking functionality to the instruments within the step sequencer grid, disallowing multiple users from adding inputs to the same instrument at the same time.
- Research new possible icons to replace the wrench and the circled “x” that represent locking and unlocking a control respectively. Also add labels above, below, or beside the icons. Users should be asked what they expect the icon stands for in our next iteration of user testing, allowing us to test the recognizability of the selected icons [2].

[2] A. Harley, “Icon Usability,” Nielsen Norman Group, 2014. [Online]. Available: <https://www.nngroup.com/articles/icon-usability/>. [Accessed: 06-Jul-2020].

# Guidance System for First-Time Users (DLF) (Slide 1/2)

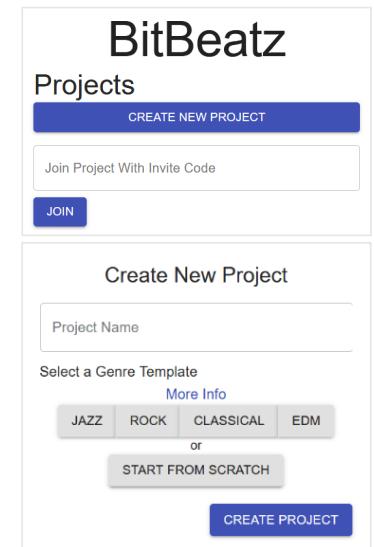
July 6, 2020  
Michael

## Insight:

A guidance system for first-time users of the application is needed to address the application's purpose, capabilities of the system, and musical terms used in the application. Users going through the creation of their first project were confused regarding what they should be doing and what the impact of their decisions would be. When presented with their new project they were once again unsure about what they could do and the meaning of many of the musical or genetic algorithm terms presented in the application.

## Evidence:

- In user testing, users understood that they were going to be making music based on the create project screen, but did not know what they would actually get out of using the app, and expressed that they were confused.
- Some users did not know the meaning of the different percussion noise types that were listed for each row of the note grid
- Users without musical knowledge showed little confidence with understanding the different musical controls.
- All users did not understand the meaning of the genetic algorithm controls, and after explained by the team stated the app should “introduce the GA concept.”
- The ‘generate percussion’ button’s purpose was not clear for users before the GA concept was introduced by the team



# Guidance System for First-Time Users (DLF) (Slide 2/2)

Recommended Action:

Whole team:

- “Decide what message [we] want to tell [our] users, but focus on inspiring them on how using [our] app will make a difference to their lives” [1]. The team needs to figure out how to concisely explain the app and its unique functionality to users in order to inspire them to create an account and use it.

Front-end team:

- Implement the user introduction strategy the team decides on into the app.
- “Get users to the most valuable part of their experience — actually using [our] app, rather than slowing them down with unnecessary roadblocks” [2]. The front-end team needs to figure out and implement a system for gently guiding first-time users through the most important parts of the music creation process of the app.

[1] D. Campbell, "Introducing your app, the right way", UX Collective, 2019. [Online]. Available: <https://uxdesign.cc/introducing-your-app-the-right-way-bd20ceb0d607>. [Accessed: 06- Jul- 2020]

[2] D. Zheng, "How to Grab Attention With Your Website Homepage Introduction", The Daily Egg, 2020. [Online]. Available: <https://www.crazyegg.com/blog/grab-attention-website-home-page-introduction/>. [Accessed: 06- Jul- 2020]

# Tutorials/Tooltips DLF

- **Insight:**

Users found it difficult to understand the functionality in BitBeatz without additional context or explanation of its features. The genetic algorithm elements were especially difficult to understand by the test users.

- **Evidence:**

During each of the user testing sessions, features had to be explained to aid in the user's understanding of the application. Key features such as the locking mechanisms and random variation slider were highlighted. The effect of modifying random variation was deemed unintuitive and, furthermore, the locking feature was expressed as "useless" by one of the test users before its significance was shown during a simulated collaboration. In terms of the locking mechanisms, as expressed in an article from The Center for Information-Development Management, "the proliferation of icons steepened the learning curve to Everest-like proportions because most of them represented the commands they triggered in ways that were not obvious to new users" [2]. Suggestions by users included exploring tooltips, a help section, or a tutorial for first time users.

- **Recommended Action:**

Entire Team: Draft and implement tooltips for all relevant components and test whether users prefer a version of BitBeatz with an initial tutorial or one with help icons for various features.

# UI Colors DLF

- **Insight:**

Within BitBeatz' interface, test users indicated certain aspects would be more appealing with different colors and that the locking mechanisms' icons were confusing.

- **Evidence:**

Users indicated that they disliked the contrast between the red color in the beats section and the blue color used in the rest of the interface. Additionally, they found that the use of the “X” icon, as seen above in Figure 4, was unintuitive for confirming their edits and rather assumed it would discard changes. It is discernable that this choice of color and symbol was incorrect as “people perform an assignment color inference process when they interpret color-coding systems” and often associate red with a warning or caution [3].

- **Recommended Action:**

Front-End Team: In the settings section, change the “X” button’s functionality to discard changes and add a checkmark button for confirming edits. Also determine the ideal, consistent color scheme through a combination of user testing and use of accessibility tools to ensure WCAG compliance



# Check-in with Nolan

- Nolan tested our website and found a lot of the same issues we found during user testing
- Bad location for play button, wrench isn't obvious, random variation is confusing. Also new insight: chat send icon looks like a share icon used for email or something
- Asked if we'll have more slider controls. This is something we need to discuss in our next iteration.

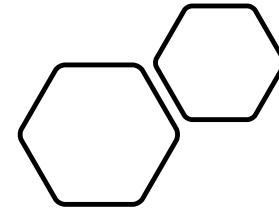
July 12th  
Kevin

# Prototype

Rudimentary GA functionality implemented in python.

- Basic Percussion Class developed
- Combination function is buggy, going to attempt uniform crossover

# Third Design Iteration



# Third Iteration Planning

- Decided we should focus our initial effort for our next development cycle on fixing the UI issues that were discovered in testing, connecting the backend and frontend, and diversifying our GA to stop it from plateauing so fast.
- Griffin, Michael, Thomas and Peter will fix the UI. Michael will make tickets and they can be claimed on Trello. Kevin and Greg will investigate fixing the GA.
- Investigate sound APIs or creating our own sound API/cloud function. Peter will look into this.
- We need to work on making these improvements over the next few days.
- We need to ask Prof. Borland for a code review. Possibly for Monday?

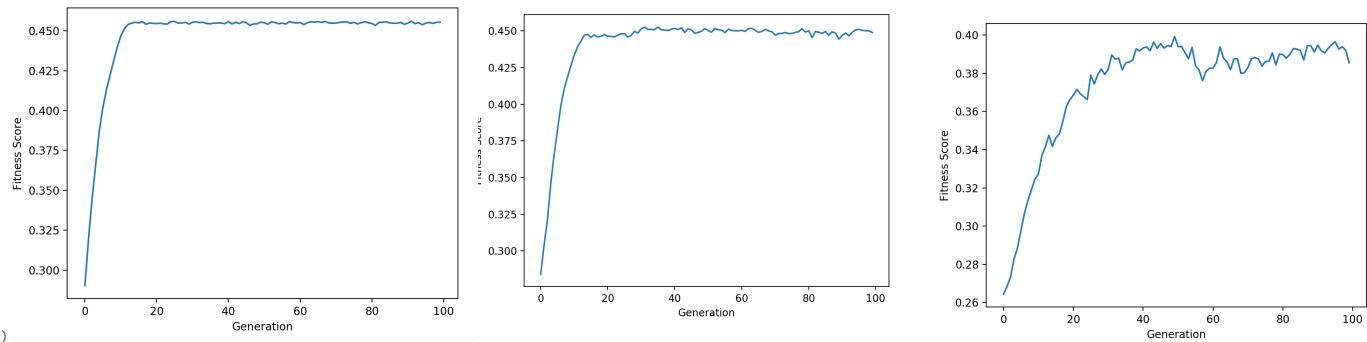
# Prototype – GA modification, selection process

- As was investigated earlier, the GA was not producing enough variation in the generations onward, and eliminating all interesting traits far too early
- A modified selection approach was implemented and tested. The selection approach prior only took the top 50% of the scores to produce the next generation.
- A user specified impurity constant was added to allow a certain amount of less fit classes to survive and combine, which removed the issue of early convergence

```
def create_next_gen(pop, ideal, impurity = 23 ):
    #add a variation variable
    #gets the top fit of the population
    scores = get_scores(pop, ideal)
    scores = sorted(scores, key=lambda x: x[1])
    mates = scores[-int(len(scores)/2):]
    lucky = scores[:impurity]
    mates = lucky+mates[impurity:]
    next_gen=[]

    for i in range(4):
        #shuffles parents four times to make kids
        shuffle(mates)
        moms = mates [-int(len(mates)/2):]
        dads = mates [0:int(len(mates)/2)]
        for i in range(0,len(moms)):
            next_gen.append(combine(moms[i][2],dads[i][2]))

    return next_gen
```



Increasing impurity (from left to right)

July 17-18, 2020  
Griffin

# Prototype – Clean up from last iteration and some new features

July 17<sup>th</sup>:

- Removed the toggle switch and controls component that had to do with the prototype with locks. It was found in testing that the locks were good.
- Added tooltips to disabled sliders to give the user more information on them being locked.
- Changed the lock icons from wrenches and x's to actual locks, both unlocked and locked depending on the state.
- Fixed the truncation of usernames based off their emails.
- Moved video cams to the bottom of the screen. Show multiple at a time to reflect the feedback from our user testing.

July 18<sup>th</sup>:

- Added the locks to the grid similar to how they were implemented in the controls.
- Added collaborator avatars to the locks on the grid to show when others were making modifications.

Merge branch 'master' of github.com:Bitbeatz/bitbeatz	a33fef0
video cam adjustment	8311f04
Merge branch 'master' of github.com:Bitbeatz/bitbeatz	eb590b5
removed second prototype toggle, added tooltips to disabled sliders, ...	7953b53

added collaborator avatars to locks on grid	c6bd4e9
Merge branch 'master' of github.com:Bitbeatz/bitbeatz	d3e3744
locks work without collab for grid	7c311c5

July 17-18  
Michael

# Prototype

Front-end / UI work for the 3rd iteration

## Design a product introduction

in list Done ☺

MEMBERS LABELS DUE DATE

ML +

Design +

+ Jul 18 at 2:01 PM COMPLETE ▾

### Description Edit

From DLF - need to give users context of what the product is when they first load the page (before they sign up/log in)

## Pretty up the login screen

in list Done ☺

MEMBERS LABELS DUE DATE

ML +

1 +

+ Jul 18 at 2:01 PM COMPLETE ▾

### Description Edit

The login screen is the first things users see; it shouldn't look this bad  
maybe make signup/login into separate components

\* See June 27 slide "Prototype Work – Michael" for ticket management methodology using Trello

## Fix the sidebar

in list Done ☺

MEMBERS LABELS

ML +

1 +

### Description Edit

Should use icons when navbar is collapsed  
The active project should show up differently  
Home button needs to be centered

## Chat fixes

in list Done ☺

MEMBERS LABELS DUE DATE

ML +

1 +

+ Jul 20 at 2:01 PM COMPLETE ▾

### Description Edit

Empty messages can be entered  
The chat doesn't scroll, just extends vertically forever

July 20-22, 2020  
Griffin

# Prototype – Mostly UX fixes, wording, labels, styling

July 20<sup>th</sup>:

- Fixed some styling on the controls and avatars

July 21<sup>st</sup>:

- Modified the intro text blurb to be less wordy.
- Synced up all the colours across the app. We had slight variations in the reds and so on before.
- Added more tooltips to the project page, including the locks, control labels, play button.
- Modified the random variation control to have its max at 40% to ensure it matched up with the GA.

July 22<sup>nd</sup>:

- Added grid labels for the beats (ie. 1,2,3, 2,2,3, 3,2,3, 4,2,3). Made them dynamic so that they show and disappear as the grid grows and shrinks.
- Added a general explanation tooltip with Greg's blurb about modifying the grid and clicking generate.

shrunk padding on controls		3fe0f62		
griffincook committed 16 days ago ✓				
Merge branch 'master' of github.com:Bitbeatz/bitbeatz		2ae8da2		
griffincook committed 16 days ago				
avatar css		25240fa		
griffincook committed 16 days ago				

Merge branch 'master' of github.com:Bitbeatz/bitbeatz		75c4a9d		
griffincook committed 14 days ago ✓				
added explanation tooltip		8f26aa5		
griffincook committed 14 days ago				
expanded grid labels		a150674		
griffincook committed 14 days ago ✓				
Merge branch 'master' of github.com:Bitbeatz/bitbeatz		5a2907f		
griffincook committed 14 days ago ✓				
quick hard code fix for grid labels		40b8176		
griffincook committed 14 days ago				
capped rando var at 40		a0d8343		
griffincook committed 15 days ago ✓				
Merge branch 'master' of github.com:Bitbeatz/bitbeatz		01ee919		
griffincook committed 15 days ago ✓				
added tooltips to lots of the project page		6c70076		
griffincook committed 15 days ago				
synced up colours		bbae5bb		
griffincook committed 15 days ago ✓				
Merge branch 'master' of github.com:Bitbeatz/bitbeatz		4ff54c7		
griffincook committed 15 days ago ✓				
modified intro blurb		17da4d0		
griffincook committed 15 days ago				

July 21-22  
Michael

# Prototype

Work for connecting the front-end / back-end

**Connect frontend and backend**

in list Done

MEMBERS **5**

LABELS

**Description**

Make the GA into a cloud function  
Push the output of the GA to firestore

**Add generating percussion spinner**

in list Done

MEMBERS **1**

LABELS

DUE DATE Jul 22 at 2:01 PM COMPLETE

**Description**

No idea why but the GA cloud function takes a while so a spinner beside the generate percussion button would be good to show that something's happening

\* See June 27 slide “Prototype Work – Michael” for ticket management methodology using Trello

July 21  
Peter

# Prototype – Add sounds to clickable drum labels

- Downloaded royalty free sounds for the bass, ride, hi-hat, low tom, and snare, then created a drum sounds resource in firebase storage
- Set the drum label buttons in the project UI to make a request to firebase to download the corresponding sound, and then play it when clicked
- Result is that all drum label buttons play their corresponding sound responsively

Commits on Jul 21, 2020

Add sounds to drum label buttons

 peterdmarshall committed 15 days ago ✓

<input type="checkbox"/>	bass.wav
<input type="checkbox"/>	hi_hat.wav
<input type="checkbox"/>	low_tom.wav
<input type="checkbox"/>	ride.wav
<input type="checkbox"/>	snare.wav

# Testing on BitBeatz.ca (User 1: 20F)

July 22, 2020  
Everyone  
Notes by Griffin

## Prior Knowledge:

- Limited digital music creation experience. Has played around with Garageband.
- Good amount of physical music creation experience. Singing all life, flute for four years, learning guitar.
- Online collaboration platform experience has come from this class, powerpoint, mirro.

## Using the App:

- Doesn't want to create an account, doesn't like giving away email.
- Thinks the choices for genre are odd.
- Understands grid time labels but thinks a beginner might get confused.
- Tooltips helped her navigation of the app a lot (ie saw the tooltip saying the control was locked and moved to the lock).
- Likes the percentage on the random variation slider.
- Didn't know who she was talking to in the chat, wants longer name or way to tell. Thought Griffin was Greg since we only show G.
- Thought the grid and controls were unrelated.
- Doesn't like how the grid gets overwritten by the GA without any ability to go back or confirm the change.
- Didn't know to click the instrument sounds.

## Post Questions:

- What did you like most? -- Liked share code and collaboration features a lot. The sliders were good.
- What did you dislike or find confusing? -- Didn't know who was who in the chat and when they were controlling things. Wants more control over who can join her project, like how Zoom allows you to accept someone into the room.

# Testing on BitBeatz.ca (User 1: 20F)

July 22, 2020  
Everyone  
Notes by Michael

Notes:

- Doesn't like being asked to create an account
- Said the selection of genres was odd
- Need to better explain the grid timings
- Really need tooltips or colors for the user avatar icons
- Understood locking well
- Didn't get the connection between the grid and controls
- Need explanations for most things still
- Wasn't clear when people were joining her project
- Wants to see "name" has joined the project notifications – make a ticket for this
- Webcams should be much clearer
- Webcams should be optional – take up too much space and not always wanted

# Testing on BitBeatz.ca (User 1: 20F)

July 22, 2020  
Everyone  
Notes by Peter

- Prior experience with music: grade 8 piano
- No experience with digital music
- Didn't like having so many locks on the UI, felt it was restrictive
- Grasped the grid UI relatively intuitively, the beat labels acted as a signifier
- Mentioned that having both text and video chat could be redundant
- Felt like there was a lack of capability for iteration after the first generation of the GA

# Testing on BitBeatz.ca (User 2: 21F)

July 22, 2020  
Everyone  
Notes by Griffin

## Prior Knowledge:

- Limited digital music creation experience. All from 361.
- Physical music creation: Grade 8 piano, hasn't played for a while though.
- Online collaboration platform experience has come from this class, and school in general.

## Using the App:

- Didn't have matching passwords when creating an account and found this annoying. (Maybe make the option for them to be visible)
- Didn't initially realize she had control over what was locked. Took about 5 seconds to understand.
- The tooltips really helped her understand the application more.
- Thought the webcams at the bottom were advertisements.
- Didn't notice that she could click the drum sounds.

## Post Questions:

- What did you like most? -- Appreciates the chat for collaboration.
- What did you dislike or find confusing? -- Hard to tell what changed after collaborators were done making modifications. Its hard to iteratively make music since the GA just overwrites your work and you're "back at square one".

# Testing on BitBeatz.ca (User 2: 21F)

July 22, 2020  
Everyone  
Notes by Michael

## Notes:

- More info on create project screen is not clickable as expected
- Thought the webcams were ads – placeholder could be better
- Just general confusion – really need better explanations for things
- Likes concept of having a history component, felt the chat filled this role alright but could have something dedicated to this – make a ticket for this
- Didn't notice the drum sound buttons – need to look more like buttons
- Would like a change notification for the controls and grid – can take from Sprint prototype
- Feels like the iteration of the music is important and would be a good future step – related to history component?

# Testing on BitBeatz.ca (User 3: 21M)

July 22, 2020  
Everyone  
Notes by Griffin

I acted as the interviewer for this interview so the notes may be missing a couple things.

## Prior Knowledge:

- Digital music creation: more so recently, in between a beginner and intermediate.
- Physical music creation: Plays guitar.
- Online collaboration platform experience has come from this class, and school in general.

## Using the App:

- Likes having the choice of genres since it allows users to have a good start without requiring a ton of knowledge.
- Saw the info icon right away and read it to understand the platform.
- Clicked the GENERATE button and once it was done loading thought he might have done something wrong since he didn't immediately see the change in the grid.
- Would like a volume/velocity control.
- Starting with the list of instruments from the genre is good but he would like to be able to add and subtract them.
- Having the percentages on the random variation slider tooltip and "some, more, most" as labels is good but some explanation as to why 40% is most could be good.
- Grid labels were confusing. Maybe it's just because we couldn't play the song at this stage of our prototype.
- Prefers voice over video.

## Post Questions:

- What did you like most? -- Coolest thing is the live chat, and really likes the indications of where users are with the colours and icons. The general layout is good.
- What did you dislike or find confusing? -- Grid labels could be clearer. Position of webcams isn't ideal.

# Testing on BitBeatz.ca (User 3: 21M)

July 22, 2020  
Everyone  
Notes by Michael

Notes:

- No issues with product intro / create project flow
- General intrigue, impressed with cleanliness of the app
- Missing volume setting – make a ticket
- Wants to be able to add sounds / remove – custom instrument set?
- Random variation still not intuitive
- Timings not intuitive still
- Not interested in video, but wants voice – need ticket to toggle video
- Liked live collab
- Didn't see the sidebar – should be extended at first?
- Note velocity?

July 24, 2020  
Griffin

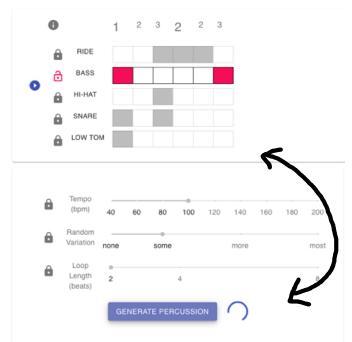
# Disconnect Between GA and Grid DLF (slide 1/2)

## Insight:

The intended connection between the step sequencer grid and the genetic algorithm is neither clear nor functionally ideal. It is hard for users to iterate upon their beat creations after running the genetic algorithm which updates the grid. Updates to the grid can also be difficult to notice and lack visual cues.

## Evidence:

- User interviews were conducted with three users, Jenna, Sadie, and John (users have been given aliases), each lasting approximately twenty minutes. All the users had previously studied music and played instruments, and John had recently been learning how to create digital music as a hobby. They were all asked to use Bitbeatz while speaking their thoughts aloud. All the users modified the step sequencer grid before clicking the “Generate Beat” button to run the GA.
- Jenna and Sadie both vocalized a dislike in their lack of control to confirm the modifications from the genetic algorithm.
- Sadie mentioned that she would appreciate to be able to go back to previous versions of her beat after clicking generate. She said she felt as though she was going “back to square one” every time she clicked generate and felt as though it would be very difficult to make iterative changes to her beat.
- Jenna stated that she fully did not realize there was a relationship between the grid and the controls in the lower part of the screen until we told her in a later portion of the interview.
- John said that he thought he did something wrong after clicking the button to run the genetic algorithm because he did not notice the grid update. Once we alerted him of the updated grid, he understood its functionality. One of the 10 Usability Heuristics for User Interface Design from Nielsen Norman Group is that “The system should always keep users informed about what is going on, through appropriate feedback within reasonable time” [2]. Our application is missing that feedback that can alert a user of the changes.



[2] J. Nielsen, “10 Heuristics for User Interface Design,” *Nielsen Norman Group*, Apr. 24, 1994. [Online]. Available: <https://www.nngroup.com/articles/ten-usability-heuristics/>. [Accessed: Jul. 22, 2020].

# Disconnect Between GA and Grid DLF (slide 2/2)

## **Recommended Action:**

### **Web Dev Team:**

- Initially show the newly generated beat in a location other than the main grid and give the user the options to confirm the new beat, generate a new one, or cancel the generation process. This could be implemented in a modal that could pop over the screen after the “Generate Percussion” button is pressed. If the user confirms the new beat, then it can replace the current one in the main grid with a temporary visual highlight cue on the entire grid.
- Store previous versions of the beat within the database and create a control on the interface to allow navigation to previous versions of the beat. In future user testing, we could observe if users use this feature or if it is overkill after adding the confirmation modal.

# Missing Musical Expression DLF (slide 1/2)

**Insight:**

Our application lacks an overall ability to modify volume for individual instruments and notes. This hinders the ability for musicians to add expression to their beats through dynamics.

**Evidence:**

- User interviews were conducted with three users, Jenna, Sadie, and John (users have been given aliases), each lasting approximately twenty minutes. John had recently been learning how to create digital music as a hobby. They were all asked to use Bitbeatz while speaking their thoughts aloud.
- John noticed that there was no way to modify the volume or pitch of the notes. The ability to do this through a velocity control was a key feature that he had become accustomed to while using other digital music creation platforms. He noted that having that extra control allows him to add more personal touches to the music and take on more ownership and pride over his creation.
- Jenna and Sadie both didn't quite know what they wanted to make for a beat and mostly clicked around the grid inputs without too much thought.
- Sadie felt a slight lack of ownership over her song.
- "Velocity is the force with which a note is played, and it is vitally important in making MIDI performances sound human" [3]. The stronger the force, the louder the volume. "With virtual drums, higher velocities will usually result in not only louder hits but also sharper transients and brighter tone, just as with real instruments" [4].

[4] macProVideoDotCom, "AudioPedia 109: MIDI - 5. Velocity," *YouTube*, Dec. 1, 2017 [Video File]. Available: <https://www.youtube.com/watch?v=84RqoQG7PkU>. [Accessed: Jul. 21, 2020]

# Missing Musical Expression DLF (slide 2/2)

## Recommended Action:

### Greg (Musical Expert):

Research/propose ideal velocities for specific beats in different genres. Place an emphasis on Jazz since that has been the focus of our prototype. Figure 8 includes more information on how dynamics are measured with velocity and their relation to vocal volume [5].

### Web Dev Team:

- Implement a control on the interface to modify the velocity of each beat. This could be done by having one slider that controls the velocity of a single selected square on the grid. The user could select a square by toggling on a “velocity mode” that changes their clicks from inputs to selecting squares.
- Implement functionality within the sound API to control the volume of individual notes. If the volume can get implemented properly then we can focus on the transients and tones, but they should not be the initial focus. Our implementation should be user tested since there is no MIDI standard for interpreting velocity [6].
- Implement a master track volume control.

### Backend Team:

- Explore modifying the genetic algorithm’s fitness function to encompass an ideal velocity based on the genre and the velocities input by the users.

[5] “Dynamics (music),” Wikipedia, 23-Jul-2020. [Online]. Available: [https://en.wikipedia.org/wiki/Dynamics\\_\(music\)](https://en.wikipedia.org/wiki/Dynamics_(music)). [Accessed: Jul. 23, 2020].

[6] R. B. Dannenberg, “The Interpretation of MIDI Velocity,” pp. 193–196, 2006.

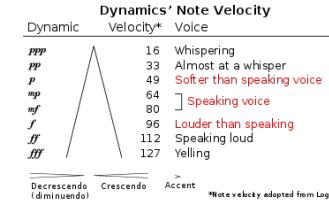


Figure 8: Visualization of velocity [5]

July 24, 2020  
Kevin

# Video DLF (slide 1/2)

**Insight:**

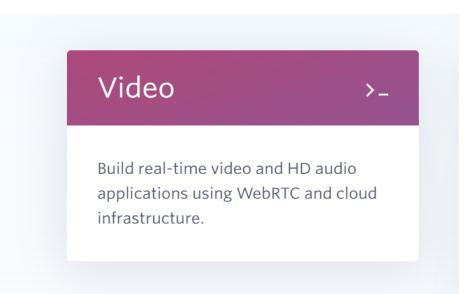
Seeing the other users via video chat would help increase user enjoyment of the BitBeatz Experience

**Evidence:**

- Users consistently mentioned that they wished that the video chat was a functional component, throughout all rounds of user testing. Reference prior using testing slides

July 24, 2020  
Kevin

# Video Chat DLF (slide 2/2)



## **Recommended Action(Kevin GA Dev):**

Expand the ability to communicate using the Twilio Video API (Doza, 2020)

## **Web Dev Team:**

Implement the preexisting API into the placeholder currently used to represent video chat. This API has preexisting React components which should make this a less intensive change to make[1].

1.Doza, T., 2020. Twilio/Twilio-Video-App-React. [online] GitHub. Available at: <https://github.com/twilio/twilio-video-app-react>

Figure 8: Visualization of velocity [5]