1. Design Options

1.1. UI Options

To increase the ease in which a user could understand and use the product, several user interface-related elements were presented in testing. The overall goal of this step was to evaluate which of these options users found helpful, and which were not necessary and/or a hinderance to the overall user experience. Each element was included in some tests and excluded in others, in order to discover the combination that would lead to the most positive user experience. Evaluation was conducted via observation, questioning, and secondary research to support final decision making. Some of the options described in this section were tested in the first and second iterations of user testing, prior to the testing described in section 2 of this report.

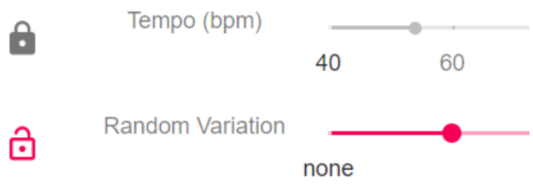


Figure 1: Locking variation sliders

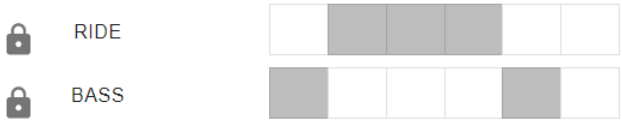


Figure 2: Locking note inputs

The first option explored was the locking feature. During the sprint, it was hypothesized that users would have difficulty trying to edit controls collaboratively, and that they would end up overwriting each other's work. As a potential solution, it was suggested that the users would be able to lock a specific element they were working on. 3 separate tests were conducted, one allowing a user to either lock both the variation sliders (Figure 1) and note inputs (Figure 2), another allowing a user to unlock only the variation sliders, and a third where users could not lock anything. As anticipated, in the case where no locks were included, users had trouble keeping track of who was building each part, and often overwrote each other's work. Most tests where locking was not available saw frustration with work being overwritten within both the note input and variation slider sections, proving the necessity of locking mechanisms in both cases.

Another tested variant was the degree of explanatory features required for users to properly use and understand the product. The goal was to have a beginner musician be able to understand the usage of every aspect of the product, while minimizing the number of text-related elements and clutter on the screen. Initial tests did not include explanations for any of the sections, in order to determine which parts were intuitive and which parts required guidance. At the end of these initial tests, users were asked if a tutorial would be beneficial in explaining all the relevant features of the product. Most users, however, responded negatively to the concept of a tutorial, describing them as excessive and unnecessary. Thus, the team opted to keep explanations concise and within the app itself, "rather than slowing … [users] down with unnecessary roadblocks at the beginning through long tutorials" [1]. By keeping the explanations within the app, we also followed the design principle of recognition rather than recall [2].

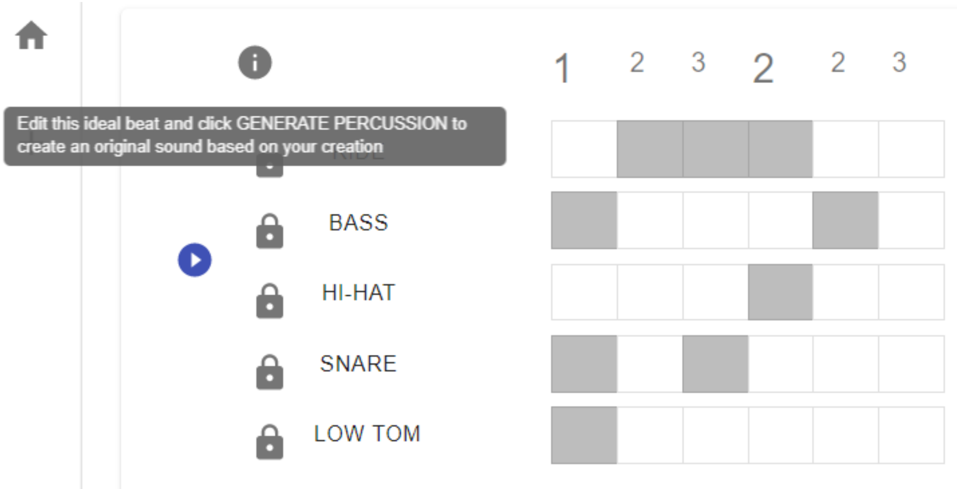

Edit this ideal beat and click GENERATE PERCUSSION to create an original sound based on your creation

Figure 3: Axis labels and info bar

Specifically, initial tests showed most users did not understand that the matrix represented the playing of different drums over a time interval. The next round of testing included one option where sounds were played with axis labels demonstrating time, and another option where an additional info bar was added as well as sounds and labels further describing the matrix's purpose (Figure 3). Most beginner users from these tests found that sounds and labels weren't enough to explain the matrix's purpose, and no users objected to the text in the info bar, originally believed to be too cluttered. For this reason, the info bar, labels, and sounds were added to the prototype.

The design options considered within the user interface were the webcam configurations. Testing in the sprint phase revealed users prefer to see their peers while writing music to better understand motivation through body language. Therefore, two different webcam styles were tested. The first had space for a single feed with the ability to toggle through different users, allowing for the feature to take up as little space as possible (Figure 4). The second had all the feeds visible at the same time, taking up more space and causing more clutter, but in return allowing for users to see everyone simultaneously (Figure 5). During testing, many users expressed frustration with the first option, explaining that they did not want to continuously toggle through different feeds and wanted to be able to focus more on the music. Conversely, the second option provided very positive results, as all users expressed interest in being able to see all feeds and no users considered it to be too cluttered.

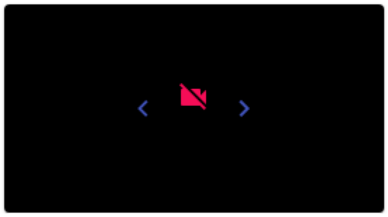


Figure 4: Toggle video option

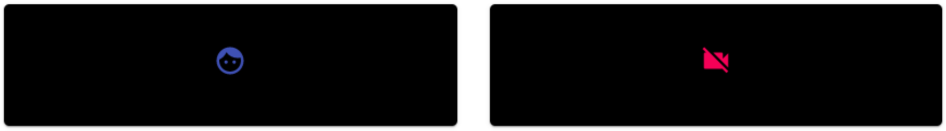


Figure 5: Multi-feed video option

1.2. GA Options

The genetic algorithm implemented was selected in a relatively straightforward manner, after the context of the UI was decided. The genetic algorithm needed to be built to breed a generation of beats that seek to emulate an ideal standard, which is initially defined by the products genre, but can be further modified by the user to add an additional level of customization. We iterated on this, drawing on materials shared in the early modules of the course.

The first thing that needed to be created was a class with which to define population members with specific chromosomes outlining how each beat was to be played. The percussion class was created to fill this need, containing a main array outlining the full chromosomes, with each row containing alleles for part of the drum kit. The class also contained scalable row and column lengths to adapt to genre. This would allow the extension of ideal, and more complex beat patterns for the user, increasing user freedom.

The fitness function to rate these population members was rather simplistically developed to evaluate each drumline of alleles against a targeted ideal created by the user in the frontend. Each drumline had a specific weighting and priority. Matching positionally would be valued differently according to each component, placing more emphasis on bass and ride, and incrementally less on the snare, hi-hat, and tom respectively.

The reproduction function will take an indeterminate number of random percussion objects, in our case one hundred, as the initial generation. The population is then scored against the fitness function, with the top fifty percent of the scores being allowed to reproduce. Depending on the level of impurity outlined by the user in the frontend, a select number of individuals with scores below the 50th percentile will be allowed to reproduce, producing more diverse beats in the next generation. The percussion objects are then paired for reproduction twice, via a random shuffle, creating the next generation of 100 percussion objects. The cycle continues for as many generations outlined by the user in the frontend. The combination function takes a pair of percussion objects and randomly selects from which parent the child will take an allele from. As it stands there is an equal chance that a child will inherit a gene from either parent. A random mutation parameter also exists to make a possible random change to a random gene during the process. The mutation chance can be altered with the mutation parameter. This combination function is implemented between each pair.

Purity and mutation variability were tweaked throughout the design process to ensure that the beats did not hang on local maxima on the fitness plane, which as a problem in the initial iterations. Graphs of overall fitness scores by generation were produced to qualitatively evaluate the most effective methods of selecting the non-fit objects for reproduction and mutation levels during combination.

2. Testing Results

A series of user tests were conducted to get users hands-on with the product. This provided insights on the quality and clarity of different components and experiences of the product. The first aspect tested was the collaborative experience that users get when creating a project with others. We wanted users to feel like collaborating can be done without jumping through hoops, and that collaboration can add to the digital music making experience, not make it more difficult. The main elements of this experience include project inviting and joining, working together on the project through the controls and grid, and general team presence on the project. The experience has a similar flow to that of Google's online collaborative tools, so it was expected that users might ask for colours to clarify who other users were, and for ways to see who was presently working on the project. The product is still a medium fidelity prototype so other shortcomings and oversights in the collaborative experience were expected, though nothing specifically. The collaborative aspect of music was also tested in earlier iterations and got positive feedback so the satisfaction with collaboration in the current design was expected to be middling to high.

Testing confirmed the expected results and provided other unexpected results. All users found the product invite flow easy, remarking that they loved how it was just a one step process. However, some users remarked that while inviting other users was easy, they wished there was some sort of notification when someone joined the project to be more informative and make the project feel more alive. Users were happy to see the first initial of their collaborators when they were modifying parts of the project, but there was no way to distinguish between users with the same initial. One user said they wanted to see a notification when other users changed controls or the grid, an idea that was first seen in one of the Sprint solutions. Overall, all users said they enjoyed the collaboration, and the suggested changes are all low risk implementations that can be tested with the final iteration.

The next item that was tested was the communication channels available with which users could emulate the in-person music making experience as a team. This included a project chat window, the idea of having voice chat, and video chat placeholder windows. Testing was done with the intent to find out which channels users found most useful so these could be focused on in the next iteration. From earlier testing with non-functional prototypes, it was expected that users would favour audio and video communication but still value the chat.

When testing, all users expressed interest in the video feature, but some said they would also like to have it displayed optionally as they felt it took up a considerable part of the screen that they found was more useful to dedicate to other components. All users expressed a desire for fully functional voice chat as expected. The text chat received better feedback than expected, with all users saying they enjoyed it. One user expressed that they also liked the context it could provide when joining new projects that were already in progress. It is unclear if an alternate *history* component would be of more use for this purpose and is also something to consider for future work. The results suggest that all the channels are needed and should be further implemented, but that the option to hide each one may be useful to increase screen space when they are unused.

The note grid was another major item that was tested for its usability and intuitiveness. The grid is composed of a set of rows for each instrument that combine to represent a percussion beat for the genre. Testing was performed to see if users would correctly figure out what the function of the grid was, what the timings for the beats listed above the grid rows were, and the meaning of the instruments. It was expected that users would associate the grid with the music they were creating, if not immediately, then with the aid of the information tooltip beside the grid. We also expected that most users would understand the timing system that appears over the grid for each note.

Testing showed that the grid was less intuitive than expected. Most users recognized that the grid corresponded to the music they were creating but were not confident in this and felt like this could be expressed more clearly. Users did not understand what the timings above the grid meant, although some of this confusion may have come from testing with jazz which typically uses triplets. Regardless, the clarity of these timings should be improved in future work. Users did not realize that the instrument labels for each row could be clicked to play the sound corresponding to the instrument, and it is clear from this that the buttons should be better represented. The changes required to address these problems with the grid will require a medium amount of design and implementation.

Quantitative testing was done throughout the process of working on the GA by graphing the average generational fitness scores. The initial version was found to converge far too quickly on the ideal beat the user provided. This can be seen in the lower left figure. With further analysis of the beat composition, it was discovered that after the first few generations only near-perfect replicas of the ideal existed in the population, with a small number of other similar beats with minor mutations. Increasing mutation probability did not seem to rectify the issue of the rapid convergence, and only increased the amplitude of the minor oscillating behavior seen in the average generational fitness score of the later generations.

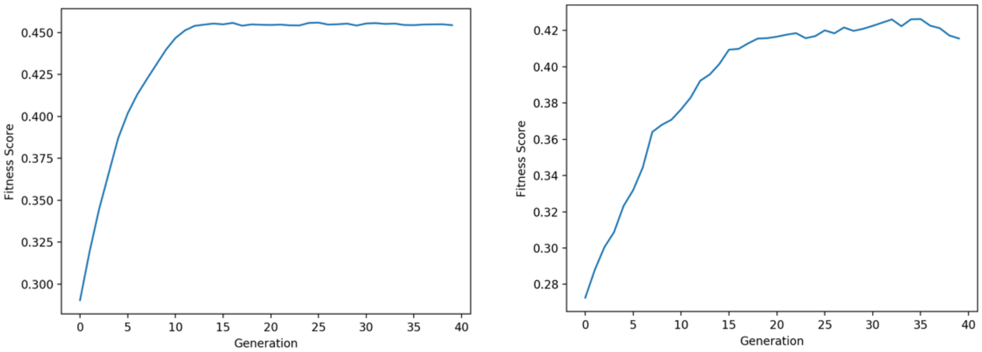


Figure 6: High selective pressure (left) and lower selective pressure (right)

After seeing this, an attempt was made to reduce the rapidity by changing the reproductive process by allowing a select few of the less fit percussion objects the chance to reproduce. We tested the effectiveness of this reduction of selective pressure, and evaluated it graphically, noticing a more drawn out progression – as seen in the upper right figure.

3. Prioritized Insights + Recommendations

3.1. Disconnect Between GA and Grid

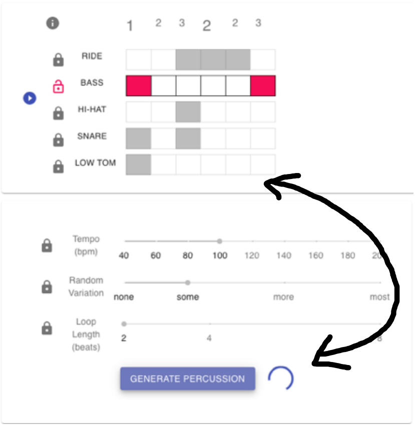


Figure 7: Prototype grid and controls

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 it's 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.

Recommended Action:
Web Dev Team:

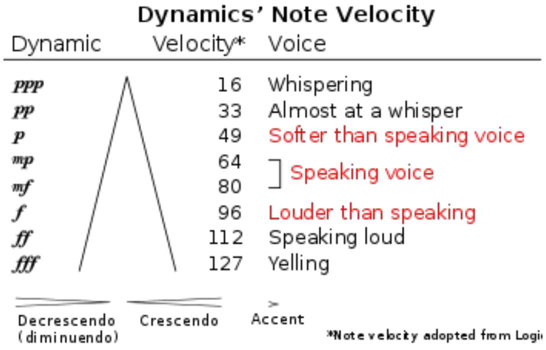
- Initially show the newly generated beat in a location other than the main grid and give the user the options to confirm the new beat, generate a new one, or cancel the generation process. This could be implemented in a modal that could pop over the screen after the "Generate Percussion" button is pressed. 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.

3.2. Missing Musical Expression

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].



Dynamics' Note Velocity

Dynamic	Velocity*	Voice
ppp	16	Whispering
pp	33	Almost at a whisper
p	49	Softer than speaking voice
mp	64	Speaking voice
mf	80	
f	96	Louder than speaking
ff	112	Speaking loud
fff	127	Yelling

Decrescendo (diminuendo)
Crescendo
Accent

*Note velocity adopted from Logi

Figure 8: Visualization of velocity [5]

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.

4. References

[1] J. Ngai, “5 Best Practices for User Onboarding,” *Medium*, Apr. 15, 2020 [Online]. Available: https://uxdesign.cc/5-best-practices-for-user-onboarding-1cc870c2ab49. [Accessed: Jul. 22, 2020].
[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].
[3] H. Jones, “MIDI Explained: What Is Velocity?” *Ask.Audio*, Dec. 5, 2017. [Online]. Available: https://ask.audio/articles/midi-velocity-what-it-is-how-it-works. [Accessed: Jul. 23, 2020].
[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]
[5] “Dynamics (music),” Wikipedia, 23-Jul-2020. [Online]. Available: https://en.wikipedia.org/wiki/Dynamics_(music). [Accessed: Jul. 23, 2020].
[6] R. B. Dannenberg, “The Interpretation of MIDI Velocity,” pp. 193–196, 2006.

5. Appendix

A. User Testing Jot Notes

User 1 (Jenna):

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 musical beginner might get confused.
- Tooltips helped her navigation of the app a lot (i.e. 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 names 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 she could click the instrument labels to demo the sounds.

Post Questions:

- What did you like most? -- Liked the share code and the collaboration features a lot. The sliders were good.
- What did you dislike or find confusing? -- Didn't know who was in the chat with her 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.

User 2 (Sadie):

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 361, 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. It’s hard to iteratively make music since the GA just overwrites your work and you're "back at square one". Feels slightly like GA made the song, less ownership.

User 3 (John):

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.

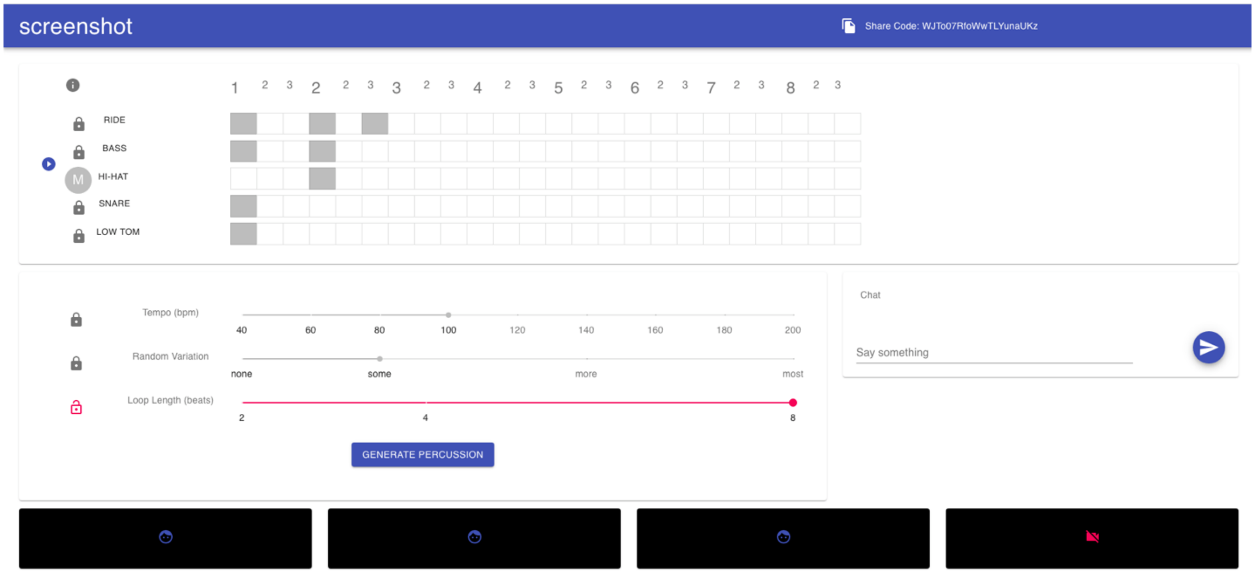
Using the App:

- 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 really 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.

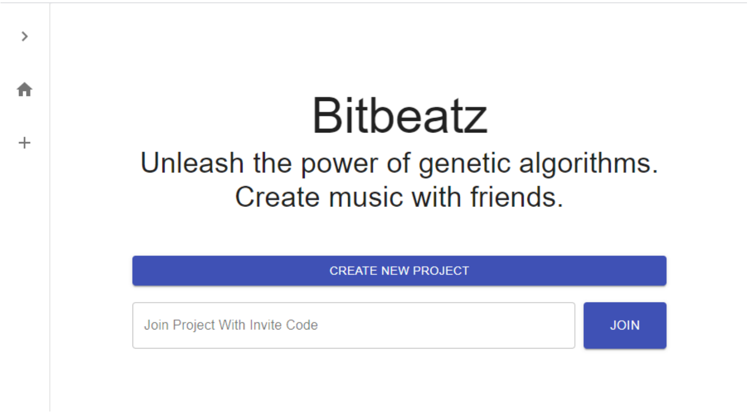
Post Questions:
- What did you like most? -- Coolest thing is the live chat, and really likes the indications of where users are with the colours and icons. The general layout is good.
- What did you dislike or find confusing? -- Grid labels could be clearer. Position of webcams isn't ideal.

B. Screenshot of the entire Bitbeatz project screen



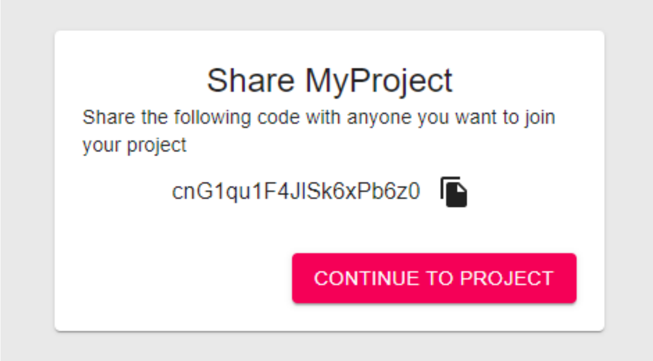
Share Code: WJTo07RfoWwTLYunaUKz

Screenshot of the BitBeatz home screen and share project screens



Bitbeatz

Unleash the power of genetic algorithms.
Create music with friends.



Share MyProject

Share the following code with anyone you want to join your project

cnG1qu1F4JISk6xPb6z0