Google Summer of Code 2020

Sugar Labs

Music Blocks Scale Degree vs n^th Modal Pitch

BASIC INFORMATION:

1. Personal Information

- Name: Aviral
- Github: aviral243
- Email: aviralgangwar24@gmail.com
- Resume: https://bit.ly/aviralGangwar
- Phone No. +91-8958539517
- IRC Nickname: aviral_243
- Location: Roorkee, India 🇮🇳
- Time Zone: India (UTC +5:30)
- Linkedin: Aviral Gangwar

- First Language: My first language is **Hindi** and I am proficient in Speaking and Understanding **English**.

2. University Information

- University: Indian Institute of Technology, Roorkee
- Majors: Electrical Engineering
- Current: II Year (expected graduation in 2022)
- Degree: Bachelor of Technology (4 Year Program)

3. Contact and Working hours

- Reachable anytime through **email** or **contact number**.

- Typical working hours include:-
  i. UTC 0330 - 0730 hrs (IST 0900 - 1300 hrs)
  ii. UTC 1030 - 1330 hrs (IST 1600 - 2000 hrs)
  iii. UTC 1630 - 2030 hrs (IST 2200 - 0200 hrs)

CODING SKILLS:

Programming Languages

- Sound Knowledge of Javascript and it's frameworks (React, Node), Typescript, PHP, C++, HTML/CSS
- Moderate knowledge in Python, Ruby, and Solidity.
- **Web Frameworks:** React, Redux, Node, Django, Django REST, Flask
- **Libraries:** Socket.IO, Jquery, Bootstrap, Semantic UI
- **Databases**: MySQL, PostgreSQL, MongoDB
- **Utilities:** Docker+Compose, CI+CD Pipeline, Postman, Jupyter, Firebase, MongoDB Atlas

Development Environment

- **Ubuntu** 19.10
- **VSCode** as IDE supported by a range of extensions.
- **Linux Shell**
- **Chrome Dev Tools**
- **Git** for version control

Apart from the technologies listed above I have sound knowledge of **Object-Oriented Programming**, architectures like **MVC**, **MTV** and **MERN/MEAN** stack.

My other interests include **Information Security**, **Blockchain**, **DevOps** and **Competitive Programming**.

ABOUT ME:

I am a 20-year-old sophomore, currently enrolled under the **Electrical Engineering department at IIT Roorkee**. I was introduced to programming concepts during High school. I developed a passion for web development in my first semester and since then most of my time went into learning new technologies and developing software using them. I have **been contributing to open-source regularly since about five-six months** now.

I have experience working closely with a team as I have been an active member of the Web Development team of E-Cell IIT Roorkee and Students' Club IIT Roorkee.

My projects:

- **Chat-App**
  - A real-time chat app. Users enter their username and the room they want to join. Users in the same room can **view and send messages to each other**.
  - It uses **Socket.IO** along with a **Node.js** backend to enable **real-time, bi-directional communication**.

- **DevConnector**
  - A social app to connect Developers. **Users can register and view other users, post online, like and comment**, and have a look at other users' public Github repositories.
  - Uses the **MERN[Mongo-Express-React-Node] stack** for implementation.

- **Utthan Foundation**

- Utthan Foundation is a **trust for rural downtrodden women**. It empowers them by providing self-employment options.
- The website's **frontend comprises** ReactJS and Sass for styling. It also features a custom Webpack **written from scratch** and Google Sheets API**.**

I am not listing all the projects here. My other projects can be viewed on my Github profile.

Open Source Contributions:

- ### Sugar Labs

I have been **working with Sugar Labs for the past 2 months**. The statistics of my contribution (at the time of writing) to **Music Blocks** are as follows:

- **Pull Requests:** 19 (**16 merged**, 1 closed and 2 open)
- **Commits:** 51 (1633++, 770--)
- **Issues:** 8 (7 closed and 1 open)

Pull requests:

- **#2061(merged):** Not setting the display to none on duplicating
- **#2071(merged):** Music Blocks doesn't go into infinite loading animation on dropping files
- **#2078(merged) :** Removed ESLint file
- **#2079(closed):** Documentation update after block addition code refactoring
- **#2080(merged):** Documentation update after block addition code refactoring
- **#2083(merged):** Migrating help strings from turtledef.js
- **#2088(merged):** Disable Keyboard input while MB loads
- **#2094(open):** Change mode to beginner/advanced without refresh
- **#2107(open):** Redo button implemented
- **#2109(merged):** Error message displaying on decreasing note value below 0

- **#2111(merged):** Fix help modal position
- **#2120(merged):** Implemented browser check to prevent MB from stopping on Firefox
- **#2122(merged):** Hardwired position parameters for help div
- **#2138(merged):** fix official app repository link in plugin readme
- **#2146(merged):** subtract one block added
- **#2149(merged):** Fix subtract block SVG
- **#2158(merged):** Conditional rendering of like button tooltip in planet
- **#2163(merged):** Feature: Cycling through block help after introductory help
- **#2171(status is closed but changes were merged):** Added find button and advanced block indication

Issues:

- **#2077:** Browser shows audio play symbol even after hitting the stop button.
- **#2086:** Documentation changes after #2082 got resolved
- **#2087:** Music Blocks not opening due to null value in voicename block.
- **#2091:** Weird on hover animation of buttons
- **#2110:** Help modal gets covered by Aux menu
- **#2121:** Help div not getting centered on first time use
- **#2137:** Official App Repository link is broken inside plugins/README
- **#2157:** Conditional rendering of like button tooltip in planet

Commits:

(Music Blocks **contributors statistics** for the time frame Jan 19, 2020–Mar 24, 2020)

- ## Rocket.Chat

  Rocket.chat is an **open-source team chat platform**.

  Pull Requests:

  - **#16122(merged):** Add Cloud Info to translation dictionary
  - **#16232(open):** Set x-content-type-options header to nosniff
  - **#1549(open):** Added steps in quick-start readme

- ## Open source Pull Requests as part of Hacktoberfest

  Hacktoberfest is a **month-long program to promote open source** contributions.

  Pull Requests:

  - **#32(merged):** Add blurb explaining what is React
  - **#7(open):** Minor changes to Readme
  - **#275(closed):** Add clap by seventeen
  - **#1266(closed):** Upgraded client and admin to react-scripts 3

PROJECT DETAILS:

Title: Music Blocks Scale Degree vs n^th Modal Pitch

Coding Mentors: Walter Bender, Sumit Srivastava
Assisting Mentors: Devin Ulibarri

Introduction:

Marry the functionality of math and computation with expectations of musicians, by working on **Issue 2058 - Scale Degree Design Path Proposal**.

Synopsis:

There is an underlying issue with the scale degree block as we have it now. The current block does not perform the function that musicians expect when they think of scale degree. Instead, the block functions in a way that we can specify a key/mode of a pitch length, input a number and result is a pitch in the chosen key/mode.

The current block has its utility in programming and we aim to keep it as such with a modified name. A new block for scale degree needs to be added with the desired functionality.

Project Plan:

Goals:

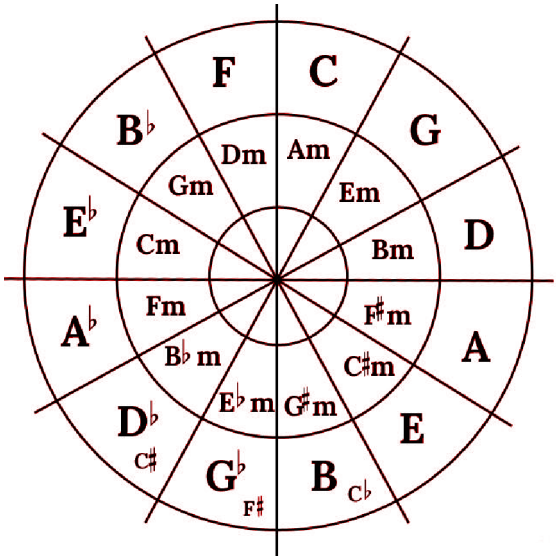
- Design and develop scale degree block in accordance with expected behavior
- Convert the existing scale degree block to a new block
  (Suggested Name: n^th modal pitch)

The Problem:

Over time as Music Blocks is being used by an increasing number of users, a share of them being Professional Musicians, some of its underlying features have been challenged. We have a series of interrelated issues here in the discussion which include:

- **#1957: Choose Key in Menu; Changes Central Options in Pitch Pie Menus**

  Music Blocks currently uses the C major scale as its default. This issue voices a concern to allow the user to choose among the 7 traditional major/minor scale options with C major or A minor being the default.
  Representation is to be carried out as using a circle of fifths:



  (Image shows a circle of fifths representing relationships among the 12 tones of the chromatic scale)

- **#2020: Scale/Mode Meta**

  Along with the issue discussed above, it proposed two more issues and hence two more blocks to be added. A key block that does nothing but offers common 7 note choices and an advanced transpositions block that is used to transpose snippets of music in a better way.

These both are underlying issues that are also somehow related to this project.

The particular issue addressed in this project is mainly confined to adding a block that realizes the true behavior of scale degree as musicians tend to think about it, but taking these issues also into consideration would provide a more wholesome development.

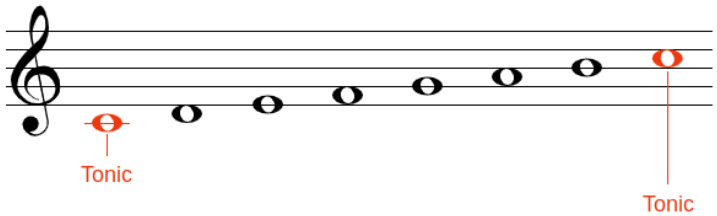
The Solution:

Since a large part of the project is still under discussion regarding design review from mentors and community there is not enough for me to discuss as part of the solution because most things will need to be figured out once we start working on it. I'm also in the learning phase as far as this issue is concerned.
I'll try to break down what has been proposed and build a few things on top of that.

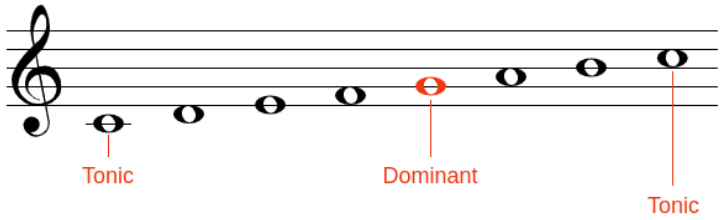
- ### What is Scale Degree?

  In simple words, each note of a scale has a special name and a number associated with it in Musical theory known as scale degree. Scale degrees are important to assess the relative distance two notes when composing music. They're also useful to indicate the size of intervals and chords and knowing whether they're major or minor.
  The naming rules are as follows:

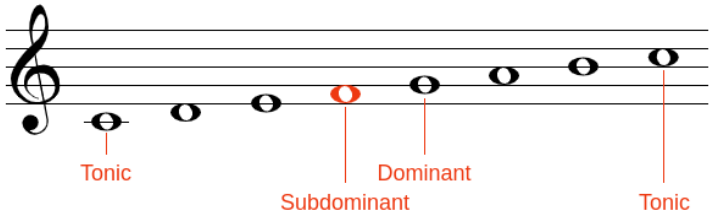
  - The first (and last) note of is called the **tonic**. The last note is sometimes also written as **tonic(octave).**



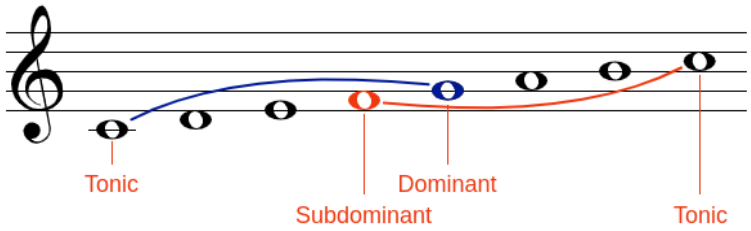
- The fifth note is called **dominant**.



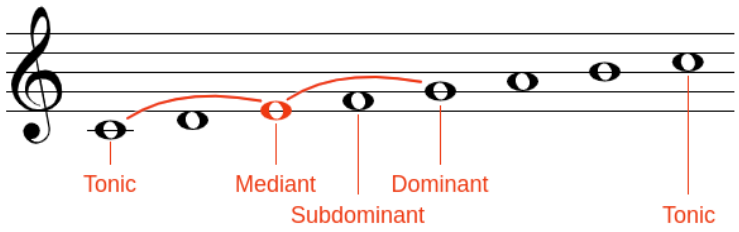
- The fourth note is called the **subdominant**. The word sub means below in Latin therefore subdominant means below the dominant.



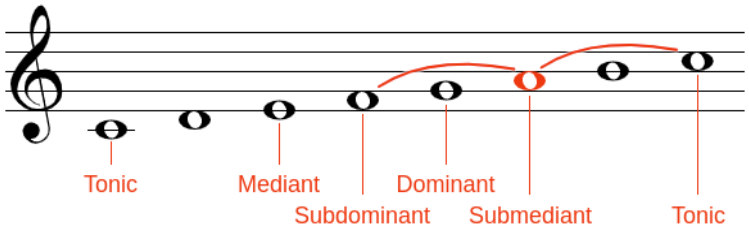
- Subdominant is the same distance below the tonic as the dominant is above it.



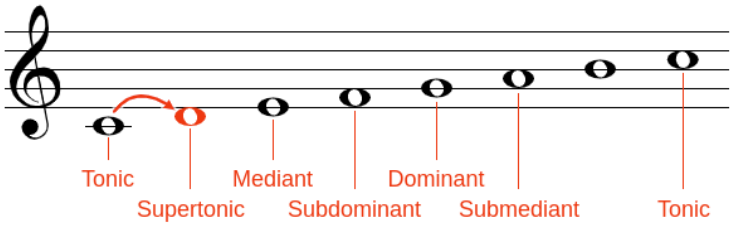
- The third note is called the **mediant** since it is in the middle of the tonic and dominant.



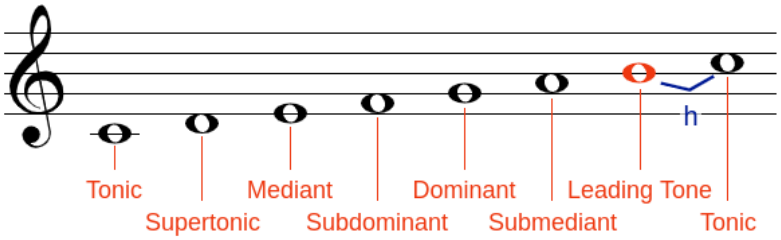
- Likewise, the sixth note is called the **submediant** since it is in the middle of the upper tonic and subdominant.



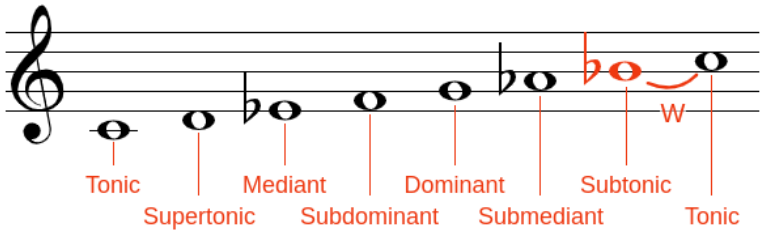
- The second note is called the **supertonic**. Super in Latin means above.



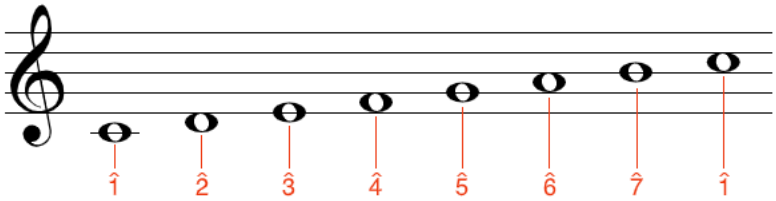
- The seventh note has a special naming convention.
  If the seventh note is half a step below the tonic, it is called a **leading tone.**



  If the seventh note is a whole step below the tonic, it is called a **subtonic.**



- Apart from these fancy names, a **number with a caret** may also be used to indicate a scale degree.



We'll mostly be using these numbers for implementation.

- Scale degree is a **movable system**. Let's understand it with an example.

  Sakura, Sakura would be sung as:
  1-, 1-, 2--, 1-, 1-, 2--, 1-, 2-, 3-, 2-, 1-, 2, 1, 6-- .....(This will end on 5)
  (7 is skipped in this snippet and 4 is skipped later)

  Now the same piece of music can also be interpreted as:
  4-, 4-, 5--, 4-, 4-, 5--, 4-, 5-, 6-, 5-, 4-, 5, 4, 2-- .....(This will end on 1)

  Notice how each scale degree is shifted by 3, playing the same musical piece.

Proposed Behaviour of Scale Degree Block:

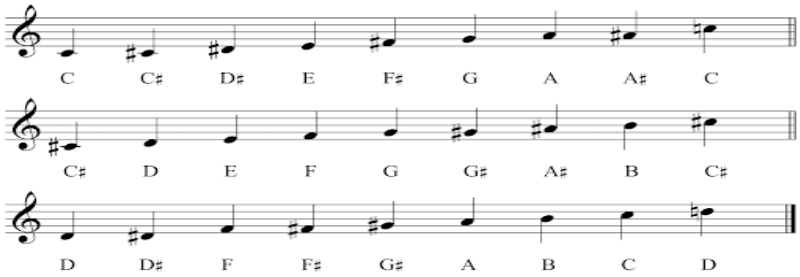
- **For 7 pitches or fewer:**

  1. Assume an underlying tonal space of 7 pitches irrespective of the key/mode selected.
  2. Scale degrees within the specified mode would be played as such.
  3. Now if the scale degree specified by the user is outside the key/mode they've selected, play a pitch based on the underlying 7 pitch system.
  4. And this pitch would be specified as a **non-scalar/non-chord tone** (utility of this is subject to design yet)

5. This is where **#2020** becomes important to resolve beforehand because which 7 note underlying system will be specified and how is subject to that issue.

- **For 8 pitches or more:**

1. Scale degree is generally not used for systems with more than 7 pitches.
2. An exception to this is a major scale that might be using a chromatic passing tone to fill the interval between two pitches.
   E.g 1, 2, 3, 4, 4#, 5, 6, 7 (4# being the passing tone in this case)
3. Not very useful to implement for diminished scales (also known as octatonic scales)



(The three diminished scales C, C#, and D)

4. In the case of such scales, we ought to show a warning and continue the previous(current) behavior of the block i.e. n^th modal pitch.

Research Source:

http://musictheory.net/

Timeline:

Since the entire project is about the implementation of a single block, the timeline should be viewed more in terms of Phases rather than weeks. Since a lot of time needs to be devoted to the ideation of the block and it's surrounding features that would account for a lot of repetitive entries in a week-by-week timeline.

Pre GSoC:

- During this time I'll try to dive deeper into musical theory especially learning more about scale degrees and their usefulness.
- Try making rough models using the acquired knowledge about how it can be implemented using JS code.

Phase 1:

- **Week 1-2 (June 1 - June 14)**
  Start with an extensive discussion with mentors and the community, especially those with a musical background about what behavior is expected from the new scale degree block.

- **Week 3 (June 14 - June 21)**
  Come to an agreement on a new name for the current scale degree block.

- **Week 4 (June 21 - June 28)**
  See if there are any changes required for the new implementation of that block. If yes, make the necessary amendments.

The logic in the above picture should be usable for the block as it is with little changes in name.

Milestone Reached: Phase 1 Evaluation

Phase 2:

- **Week 5 (June 28 - July 5)**
  Since we need to use an underlying 7 note system for the implementation of scale degree as discussed above, this phase would mostly revolve around deciding which 7 note system to use and developing that.

- **Week 6 (July 5 - July 12)**
  Discussion and work on **#1957** to decide on how to provide a key/mode selection.

- **Week 7 (July 12 - July 19)**
  Discussion and work on **#2020** to decide which underlying key/mode to provide alongside the current C major.

- **Week 8 (July 19 - July 26)**
  Put the work together and starting work on scale degree block.

Milestone Reached: Phase 2 Evaluation

Phase 3:

- **Week 9 (July 26 - August 2)**
  Code the skeleton of the block, compiling all the information gained.

- **Week 10-11 (August 2 - August 16)**
  Make necessary changes to **_playPitch()** (that basically controls the behavior of most Pitch blocks) or define a new function dedicated to scale degree block but that might not be necessary. An additional if..else condition in the existing function should be sufficient.

```
function _playPitch(args, logo, turtle, blk) {
    var useSolfegeName = false;
    if (logo.blocks.blockList[blk].name === "pitchnumber") {
        if (args.length !== 1 || args[0] == null) {
            logo.errorMsg(NOINPUTERRORMSG, blk);
            var arg0 = 7;
        } else {
            var arg0 = args[0];
        }
```

  (Existing **_playPitch()** which implements the functionality of most Pitch blocks, differentiating between them using if...else condition)

- **Week 12 (August 16 - August 23)**
  Cleanup and testing of the block mostly.

Milestone Reached: Final Evaluation

Post GSoC:

- Mostly complete remaining work on **#2020** as both issues go hand in hand. Solving it completely would mean more utility for the scale degree block.

How many hours will you spend each week on your project?

Due to the COVID-19 outbreak, my Institute has shut down unexpectedly and currently, we don't have clear notice of vacations so I can't provide the exact dates.
For what it's worth, they were scheduled to take place from 9th May 2020 - 12th July 2020.
During vacations, I can easily devote 50-55 hours per week until my college reopens and 40-45 hours per week after that. I am also free on weekends.

Other than this project, I have no other commitments or vacations planned for the summer. I shall keep my status posted to all the mentors and community members on a weekly basis and maintain transparency in this project.

If you will be off-the-grid for a few days, then mention those in the timeline.

Since I am in my sophomore year, I don't have any commitments during the period and so I will be available for almost all of the time frame. Furthermore, I do not have any planned vacations or other engagements.

How will you report progress between evaluations?

In between evaluations, I am reachable anytime through Email, IRC, Slack or a well-planned video session if required.
Further, the nature of this project is as such that I'll be continuously working on Github issues, interacting with one or more mentors. I'll let them know of the progress then and there.
I also plan to write weekly Medium articles discussing my progress with the project.
My Medium username is @aviralgangwar24. This would be my first time writing a technical blog post.

Discuss your post GSoC plans. Will you continue contributing to Sugar Labs after GSOC ends?

I do plan to keep contributing to Sugar Labs after the GSoC period because I believe that since the web is a continuously evolving place, so are its needs. Thus there will always be instances where my project will need to be worked on to have more features added to address an ever-growing populace and to cater to its needs.
Furthermore, this will also give me the opportunity to put my skills into practical use and give back to the community which I wholeheartedly want to keep doing through the platform of Sugar Labs.

Till now, the highlight of my experience with Sugar Labs has been the active involvement of the mentors. With the community growing continuously, I feel responsible for the projects I contribute to. Having picked up a lot of development skills, my major focus after GSoC would be to enhance my mentorship skills so that I can give back to this community by helping other people navigate around and hope to mentor future GSoC/GCI students.

Looking forward to contributing this summer to Music Blocks !

****