Project Title :

Music Blocks 4 Programming Framework Enhancements

Full Name : Dhruvin Kavathiya (final year undergraduate)

Email : dhruvink79@gmail.com

GitHub Username : dhruvin7k9 ,
Link : https://github.com/dhruvin7k9

Languages Known : English, Hindi

Location and Timezone: India (IST, UTC +5:30)

Skills and Experience

Programming Languages:
JavaScript/TypeScript, Python, Java, C++

Frameworks and Libraries:
MERN Stack (MongoDB, Express.js, React.js, Node.js)

Tools and Technologies:
Git/GitHub, RESTful APIs, SOAP APIs, BASH

Personal projects:

1. Competitive Programming Discussion Forum
   - Description : platform for coding enthusiasts to share knowledge, discuss ideas and write blogs.
   - GitHub Repository : https://github.com/dhruvin7k9/code-hub

2. Event Management System
   - Description : one platform for students to hassle free manage event participation and clubs to manage events related issues ,built in typescript.
   - GitHub Repository : https://github.com/dhruvin7k9/MernStack-Project-EMS

Why am I a Good Fit for This Project ?

Having worked extensively with the MERN stack and TypeScript , I have gained solid experience in building robust web applications. Also worked with Object Oriented Programming languages like Java, C# and C++, whose experience will help me with low-level design and implementation. Here's why I believe I am a good fit for the Music Blocks 4 Programming Framework Enhancements project:

1. Strong TypeScript Skills: I am proficient in TypeScript, which is the primary language used in the Music Blocks 4 project. My experience in TypeScript will enable me to understand and contribute effectively to the codebase.

2. Experience with Full-Stack Development: My experience with the MERN stack has provided me with a comprehensive understanding of web development, including backend (Node.js, Express.js), frontend (React.js), and database (MongoDB) technologies.

3. Familiarity with JavaScript Event Loop: I have a good understanding of the JavaScript Event Loop, which will be beneficial in optimizing the execution engine and handling runtime behavior in the Music Blocks 4 project.

4. Passion for Open Source: Although I have not contributed to Sugar Labs before, I am enthusiastic about contributing to open source projects. I am eager to learn and collaborate with the community to enhance the Music Blocks 4 framework.

What are you making ?

I will be working on **enhancing the programming framework of Music Blocks** (350 hours, large) specifically focusing on three key areas :

Mentors : Anindya Kundu
Assisting Mentor : Walter Bender

1. Improvements and optimizations to the Program Syntax Framework :

This involves refining the core logic and rules governing the syntax elements of the program, as well as maintaining the syntax tree representation of programs.

- Refine existing components and design considerations.
- Define business logic and rules for program syntax units.
- Maintain the Syntax Tree representing a program.
- Implement optimizations for better performance and scalability.

2. Enhancements and performance optimizations to the Execution Engine :

I will be working on building a low-overhead interpreter that supports pseudo real-time and concurrency. This includes improving the runtime behavior of programs and optimizing performance.

- Mock concurrency behavior using JavaScript Events.
- Improve runtime behavior and efficiency.
- Implement performance optimizations to reduce execution time.

3. Addition of Telemetry and Debugging support :

I will add support for monitoring the runtime states of executing programs, collecting execution information, messages, and errors, and controlling the execution process. This will involve implementing telemetry and debugging features to aid developers in analyzing and troubleshooting their programs.

- Implement telemetry features to monitor runtime states of executing programs.
- Collect execution information, messages, and errors during program execution.
- Enhance debugging support for easier troubleshooting and error handling.

Technologies will be using :

Programming Languages :

TypeScript for implementing core logic and rules, JavaScript for runtime behavior, and Node.js for server-side functionality.

Testing :

Jest for unit testing.

Version Control :

Git/GitHub for collaboration and version control.

How will it impact Sugar Labs?

The enhancements to the Music Blocks 4 programming framework will have a significant impact on Sugar Labs and its educational initiatives in the following ways:

1. Improved Learning Experience :

By optimizing the execution engine and adding debugging support, students and educators using Sugar Labs' educational tools will benefit from a more seamless and efficient learning experience. They will be able to write, execute, and debug programs more effectively, fostering a deeper understanding of programming concepts.

2. Expanded Capabilities :

The enhancements will expand the capabilities of Music Blocks, making it a more versatile and powerful tool for teaching programming concepts. This will enable Sugar Labs to reach a broader audience and cater to the diverse needs of learners across different age groups and skill levels.

3. Community Engagement :

By contributing to the development of an open-source educational platform like Music Blocks, Sugar Labs will further engage with the global community of educators, developers, and learners. The project's continued growth and improvement will encourage collaboration and knowledge-sharing among stakeholders, driving innovation in educational technology.

Timeline :

May 1 - 26 (week 0) : Community bonding period

- Set up the required environment for the Music Blocks project.
- Familiarize myself with the existing codebase and documentation.
- Discuss project requirements and goals with a mentor.

May 27 - June 9 (Week 1-2) :

- Begin implementing improvements to Program Syntax Framework.
- Work on refining core logic and rules for syntax elements.
- Define business logic and rules for program syntax units.
- Implement optimizations to improve performance.

June 10 - June 23 (Week 3-4) :

- Address any feedback or issue from initial implementation.
- Develop a low overhead interpreter supporting pseudo real-time.
- Mock concurrency behavior using javascript events.
- Optimize runtime behavior and efficiency.

June 24 - July 7 (Week 5-6) :

- Start working on optimizations to the execution engine.
- Implement low-overhead interpreter for pseudo real-time execution.
- Implement telemetry features for runtime monitoring.

Midterm Evaluation (July 8 - July 12) : Submit progress report , demonstrate completed enhancements to Program Syntax Framework and move ahead as per directed by mentors.

July 8 - July 21 (Week 7-8) :

- Continuous optimization to the execution engine.
- Integrate concurrency support and performance improvements.
- Write comprehensive unit tests using jest.
- Perform integration testing to ensure compatibility.

July 22 - August 4 (Week 9-10) :

- Begin adding telemetry and debugging support.
- Implement runtime state monitoring and error handling.
- Document new features , APIs and usage guidelines.
- Resolve any outstanding issues.

August 5 - August 18 (Week 11-12) :

- Complete implementation of telemetry and debugging features.
- Perform thorough testing and debugging of the entire framework.
- Finalize project documentation and prepare for final evaluation.
- Address any remaining issues or feedback from the mentor.

Final Evaluation (August 19 - August 26) : Submit progress report and demonstrate completed enhancements to the project.

Deliverables :

- Refinement of Program Syntax Framework.
- Enhanced Execution Engine supporting pseudo real-time.
- Addition of telemetry and debugging support.
- Comprehensive documentation covering new features and usage guidelines.

Work commitments : I will commit 25-30 hours per week to this project considering it as my only priority project for upcoming months.

Reporting Progress:
Between evaluations, I will maintain regular communication with my mentor through weekly check-ins via mailing-list email Elements . Additionally, I will provide detailed progress updates and code reviews through GitHub pull requests and issues.

Post GSoC Plans:
After GSoC ends, I plan to continue contributing to Sugar Labs and Music Blocks project. I believe that quality contributions will make this project more appealing and impact folks interested in learning music.I will remain active in the Sugar Labs community, participating in discussions, contributing code, and assisting other contributors as needed.