Project Proposal

Title

Uncover the 3x3 Rubik's Cube

Purpose

We propose the development of a web-based interface that will assist users in solving the classic 3x3 Rubik's Cube puzzle. This project will combine cutting-edge web technologies such as Three.js, React, HTML, CSS, JavaScript, and Python to create an engaging and interactive platform for Rubik's Cube enthusiasts. The primary goal of this project is to provide a user-friendly and intuitive web interface that guides users through the process of solving a 3x3 Rubik's Cube step by step. The interface will leverage the power of Three.js for 3D cube visualization, React for dynamic web components, and Python for solving algorithms.

Benefits

Interactive Learning: Users can interact with a 3D representation of the Rubik's Cube, making it easier to understand and learn cube-solving techniques through visual and hands-on experience.

Step-by-Step Guidance: The interface provides step-by-step instructions and animations, guiding users through the solving process. This helps users grasp solving methods progressively.

Accessibility: Being web-based, the interface is easily accessible to anyone with an internet connection and a web browser, allowing Rubik's Cube enthusiasts to practice and learn from anywhere.

Algorithmic Solver: The inclusion of a Python-based solver assists users by providing optimal solutions for scrambled cubes, enhancing their problem-solving skills.

Practice and Improvement: Users can generate random cube scrambles and track their progress, allowing for continuous practice and improvement.

Engagement: The combination of 3D visualization, interactivity, and user-friendly design makes learning and practicing solving techniques engaging and enjoyable.

Cross-Platform Compatibility: The interface is designed to work on various devices, including desktop computers, tablets, and mobile phones, ensuring accessibility for a wide audience.

Educational Value: It serves as an educational resource for both beginners and experienced cubers, promoting the learning and mastery of Rubik's Cube solving.

Community Building: The interface can foster a community of Rubik's Cube enthusiasts who can share their progress, compete, and collaborate in solving the cube.

Enhanced Problem-Solving Skills: Users can develop and enhance their critical thinking, problem-solving, and algorithmic thinking skills while working on solving the Rubik's Cube.

Overall, the web-based Rubik's Cube solver interface offers an accessible, engaging, and educational platform for users to learn, practice, and master the art of solving the iconic puzzle.

Major Functionality

3D Cube Visualization: Render a 3D representation of the Rubik's Cube.
Allow users to interact with the virtual cube, rotating and manipulating it in real-time.

Step-by-Step Instructions: Provide a step-by-step solving guide with detailed instructions and animations. Highlight the current step and guide users through each move required to solve the cube.

User-Friendly Interface: Create an intuitive and responsive user interface with React components.

Algorithmic Solver: Implement a Python-based solver that calculates optimal solutions for scrambled cubes. Display the solving algorithms and provide insights into the solving process.

Cross-Platform Compatibility: Ensure the interface works on various devices and web browsers.

Responsive Design: Ensure that the interface adapts to different screen sizes and orientations.

Algorithm Reference: Provide a reference section with common cube-solving algorithms. Offer a feedback mechanism for users to report issues or suggest improvements.

Target Users

Beginners: Individuals who are new to solving the Rubik's Cube and want to learn the basics and build their solving skills from scratch.

Intermediate Solvers: Users who have some experience with solving the cube but want to improve their techniques and speed.

Advanced Solvers: Experienced cubers who are looking for advanced algorithms and strategies to enhance their problem-solving skills.

Educators: Teachers and educators who want to incorporate Rubik's Cube solving into their educational programs or use the interface as a teaching tool.

Enthusiasts: People who enjoy solving the Rubik's Cube as a hobby and want to practice, compete, and connect with other enthusiasts.

Parents and Guardians: Adults who want to teach their children how to solve the Rubik's Cube or use it as an educational and recreational activity.

Competitors: Individuals participating in Rubik's Cube competitions who use the interface for practice and improvement.

Casual Users: People who simply enjoy playing with the Rubik's Cube for fun and relaxation.

Technologies

Frontend Technologies

React: React is a JavaScript library for building user interfaces. It will serve as the foundation for creating interactive and dynamic components for the interface.

Three.js: Three.js is a JavaScript library for 3D graphics and rendering. It will be used to create the 3D visualization of the Rubik's Cube, allowing users to manipulate and interact with it.

HTML: HTML is the standard markup language for creating the structure of web pages. It will be used to define the layout and content structure of the interface.

CSS: CSS will be used for styling and designing the user interface, ensuring a visually appealing and responsive design.

JavaScript: JavaScript will add interactivity to the interface, handle user actions, and implement various features.

Backend Technologies

Python: The core programming language for implementing the Rubik's Cube solving algorithm and handling backend logic.

Kociemba Library: A specialized library that leverages Kociemba's algorithm to solve the 3x3 Rubik's Cube efficiently and accurately.

Web Framework (Flask/Django): We'll utilize a web application framework like Flask or Django to manage requests, handle user interactions, and serve data from the backend to the frontend. These frameworks provide robust tools for building web applications.
Database: To securely store user progress, statistics, and related data, we'll employ a database system, ensuring data integrity and user-specific information storage.

UI Design Tools

- *Pixso*, An online UI design tool for collaborative design work using cloud.
- Graphic design software Adobe Creative Cloud for creating icons and graphics.

Project Phases

1 . Planning and Design
- Define project objectives and requirements.
- Create wireframes and UI design.
- Plan the database structure.
- Establish a project timeline and assign tasks.

2 . Frontend Development
- Develop the frontend interface using React.js.
- Implement the 3D Rubik's Cube simulation with Three.js.
- Add interactive features such as hint, scramble, and reset.

3 . Backend Development
- Implement the Rubik's Cube solving algorithm in Python.
- Set up the backend server using Flask/Django.
- Integrate the solver with the frontend interface.

4 . Testing and Quality Assurance
- Test the application for functionality and usability.
- Address any bugs or issues.
- Optimize performance and security.

5 . Deployment and User Testing
- Deploy the application to a web server.
- Conduct user testing to gather feedback.
- Make necessary adjustments based on user feedback.

6 . Documentation and Maintenance
- Create user documentation, reports, and guides.
- Maintain and update the application as needed.

By following these phases and leveraging the expertise of our team members, we aim to deliver a high-quality Rubik's Cube Solver and Simulator that meets the needs of both enthusiasts and beginners. This project showcases our skills in web development, UI design, and algorithm implementation while providing valuable tools for users.

Software Configuration Management Plan (SCM)

Team Members and Roles and Contributions

Role	Team Member	Responsibilities
UI Designers	Yuan Gao	Creating visually appealing icons and graphics.
	Chandana Nandan	Collaborating on UI design and ensuring a cohesive user experience.
Backend Developer	Yuchen Zheng	Implementing the Rubik's Cube solving algorithm, setting up the backend server, and deployment.
Frontend Developer	Xinyu Yang	Building the frontend interface using React.js, implementing the 3D Rubik's Cube simulation with Three.js, and creating interactive features like hints, random scrambles, and resets.
Project Manager	Rotational Assignment	Overseeing project progress, coordinating tasks, ensuring timelines are met, and resolving project-related issues.
Developer and QA	Shanthakumar Sivakumar	Testing the application, identifying bugs, and ensuring a smooth user experience. Assisting back-end development
UI and Document Writer	Chandana Nandan	Creating, maintaining, and organizing project documentation, including requirements, technical materials, user guides, and meeting minutes.

Project management tools

GitHub: GitHub's project boards and issue tracking will be used for version control and collaborative coding. It allows for code review, issue management, and collaboration among developers. GitHub Actions can automate tasks like testing and deployment.

Slack: Slack will serve as the primary communication and collaboration platform. Dedicated channels will be created for various project aspects, including general discussions, development updates, and bug tracking. Integrations with other tools can streamline notifications.

Google Workspace: Google Workspace (formerly G Suite) will be used for document collaboration, including requirements documents, design specs, and user guides. Google Drive will store project-related files, and Google Docs will facilitate real-time editing.

JIRA: JIRA may be considered for more extensive issue tracking and project management, especially if the project scales up or involves multiple teams. It offers advanced customization and integration options.

Zoom: Zoom will be used for virtual meetings, discussions, and team collaboration. It includes video conferencing, screen sharing, and recording features.

Visual Studio Code: Visual Studio Code, as a code editor, supports collaborative coding with extensions for Git integration and code sharing. It enables real-time collaboration on code files.

Configuration Identification

Version Control: Utilize Git as the version control system to manage the project's source code.

File Naming Conventions: Enforce consistent naming conventions for source code files, documents, and project assets.

Change Requests: Track changes through a formal change request process that includes documentation, impact assessment, and approval.

Baseline Components: Identify and establish baselines for the following components:

- Source Code
- Database Schema
- Design Documents
- User Documentation
- Configuration Control

Change Management Process: Implement a formal change management process that includes:

- Change request submission.
- Change request review and assessment.
- Approval or rejection of change requests.
- Implementation of approved changes.

Change Log: Maintain a change log to record all changes made to the project's software and associated documentation.

Version Numbering: Use semantic versioning (e.g., MAJOR.MINOR.PATCH) to track software versions.

Configuration Status Accounting

Configuration Items: Define and document all configuration items, including source code files, documentation, and design artifacts.

Status Reports: Generate regular status reports to track the current state of configuration items, including version numbers and change history.

Audit Trails: Maintain audit trails to trace changes back to their source and provide a historical record of modifications.

Release Management

Release Planning: Plan and document release schedules, including version numbers, release notes, and deployment procedures.

Deployment Process: Define and document the process for deploying new releases, including rollback procedures.

Backup and Recovery

Backup Procedures: Implement regular backup procedures for all project assets, including source code repositories, databases, and documentation.

Disaster Recovery: Develop a disaster recovery plan to ensure data integrity and availability in case of system failures or data loss.

Risk Management Plan

Risk Description	Likelihood	Impact	Mitigation Strategy	Contingency Plan
Technical challenges in implementing the solver	High	High	Conduct a thorough feasibility study and prototyping phase.	Allocate additional time for research and development.
Delays in frontend development due to complexity	Moderate	High	Use experienced front-end developers and provide training.	Extend the project timeline if necessary.
Inadequate communication within the team	Low	Moderate	Establish regular team meetings and communication channels.	Implement agile project management for better collaboration.
Loss of project documentation or codebase	Low	High	Regularly back up project files on multiple cloud platforms.	Maintain version control with Git to track changes.
User dissatisfaction with the solver's accuracy	Moderate	Moderate	Involve users in testing and gather feedback for improvements.	Develop a plan to address user concerns and iterate the solver.