**Capstone Design Project Proposal**

**Section A. Problem Statement**

The problem that our team will be focusing on this semester is helping students become more interested in the field of computer science and quantum computing. Students who may not have been exposed to computer science and programming before may not be interested in the field, and we hope that by using interactive games and block-based programming, students will become more interested. Additionally, the concept of quantum computing can seem very complex and intimidating, so having a simple user interface and game concept will help teach students about the nature of quantum problems in a manageable way. In the future, it is likely that quantum computing will continue to expand and have more applications as the technology is developed further. Because of this, helping students to become familiar with the basics of how quantum computers recognize problems and store information will help them build skills that may be applicable to their future careers.

**Section B. Engineering Design Requirements**

**B.1 Project Goals (i.e., Client Needs)**

* Create an accessible web application for K-12 students to learn quantum computing concepts.
* Develop interactive games and puzzles that teach quantum computing principles.
* Implement a user-friendly block-based programming interface.
* Enable users to program and interact with a real quantum computer.
* Provide visualization tools for quantum computer operations.

**B.2 Design Objectives**

* Increase accessibility of quantum computing education for K-12 students.
* Enhance student engagement through gamification of quantum concepts.
* Simplify quantum programming using visual block-based tools.
* Demonstrate real-world applications of quantum computing.
* Foster skills relevant to future quantum technology careers.

**B.3 Design Specifications and Constraints**

* Web-based application accessible from standard computers.
* Multiple interactive games/puzzles (including Tic-tac-toe, Connect 4, and Mancala).
* Block-based programming interface using Google's Blockly framework.
* Integration with a remote quantum computer (e.g., D-Wave) for computations.
* Visualization tools for quantum circuit diagrams and problem-solving processes.
* Game development interface using Scratch for user-created content.
* Ability to save, load, share, and switch between games,…
* Variable difficulty levels for game engines.

**B.4 Codes and Standards**

* Modular Design principles
* DRY (Don't Repeat Yourself) principle
* KISS (Keep It Simple, Stupid) principle
* YAGNI (You Aren't Gonna Need It) principle
* SOLID principles
* Python PEP 8 style guide
* Responsive Design for HTML and JavaScript
* Documentation standards (inline comments, Python Sphinx)

**Section C. Scope of Work**

**C.1 Deliverables**

* Fully functional web application with user interface
* Completed game engines for Tic-tac-toe, Connect 4, and Mancala
* Block-based programming toolbox for quantum computing
* Integration with remote quantum computer
* Visualization tools for quantum processes
* User guide and documentation

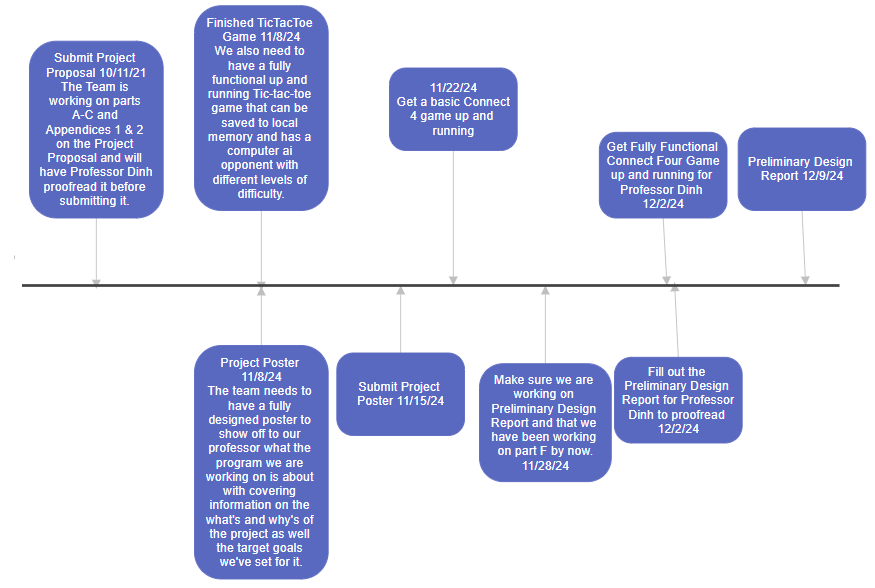
**C.2 Milestones**

* **November 8th:** Complete functionalities for existing Tic-tac-toe game
  + Integrate block-based programming interface
  + Final testing and user feedback implementation
* **November 15th:** Submit project poster
* **December 2nd:** Complete functionalities for Connect 4 game
  + Develop and integrate visualization tools
  + Project deployment and final documentation
* **December 9th:** Submit preliminary design report
* **March 10th:**
  + Complete functionalities for Mancala game
  + Documentation

**C.3 Resources**

* Google's Blockly Framework
* MIT's Scratch Blocks
* D-Wave Quantum Computer (remotely accessible)
* Front-end Development tools (HTML, CSS, JS)
* Remote Web Hosting Service
* Development IDEs (e.g., Visual Studio Code)
* Educational resources on quantum computing

**Appendix 1: Project Timeline**

**Timeline Overview:**

**Appendix 2: Team Contract (i.e., Team Organization)**

**Team Roles and Responsibilities:**

* **Project Manager:** Steven Acosta
  + Oversees project timeline, coordinates team meetings, ensures milestones are met.
* **Financial Manager:** Steven Acosta
  + Resource Management
  + Budget Management
  + Financial Reporting
* **Systems Engineer:** Robert Duncan
  + Analyze the clients needs
  + Create and integrate sub-systems
  + Develop system architecture and manage product interfaces
* **Logistics Manager:** Santiago Agudelo
  + Leads contact within and outside of organization
  + Obtains any information the team may need
  + Documents meeting minutes
  + Tracks facility and resource usage
* **Test Engineer:** Joshua Pomeroy
  + Oversees experimental design, test plan, procedures and data analysis
  + Acquires data acquisition equipment and any necessary software
  + Establishes test protocols and schedules; oversees statistical analysis of results
  + Leads presentation of experimental finding and resulting recommendations.

**Communication Protocols:**

* **Weekly Meetings:** Scheduled every Thursday at 6 to discuss progress, challenges, and upcoming tasks. We also meet on Zoom at 1 on Thursdays with our advisor
* **Communication Tools:** Utilize Discord for daily communication, Discord, Google Drive, GitHub for task management and version control.
* **Reporting:** Weekly progress reports submitted to the Project Manager for consolidation and review.

**Conflict Resolution Procedures:**

* **Open Discussion:** Address conflicts openly during team meetings to find mutually agreeable solutions.
* **Mediation:** If conflicts persist, involve a neutral third party (e.g., faculty advisor) to mediate and provide guidance.
* **Clear Documentation:** Maintain clear records of decisions and agreements to prevent misunderstandings.

**Commitment Expectations:**

* **Attendance:** Team members are expected to attend all scheduled meetings unless prior notice is given.
* **Deadlines:** Adherence to deadlines is crucial for project success; team members must communicate any potential delays promptly.
* **Quality of Work:** Deliver high-quality work that meets the project’s design specifications and standards.

**References:**

* Google's Blockly Framework Documentation: https://developers.google.com/blockly
* MIT's Scratch Blocks Documentation: https://developers.google.com/blockly
* D-Wave Quantum Computer Documentation: https://www.dwavesys.com/resources
* Python PEP 8 – Style Guide for Python Code: <https://pep8.org>
* Adobe XD Documentation: <https://helpx.adobe.com/xd/user-guide.html>
* Figma Design Tool: https://www.figma.com/resources/learn-design/
* Flask Web Framework Documentation: https://flask.palletsprojects.com/
* Django Web Framework Documentation: <https://www.djangoproject.com/>
* PostgreSQL Documentation: <https://www.postgresql.org/docs/>
* Firebase Documentation: https://firebase.google.com/docs
* SurveyMonkey User Feedback Tool: <https://www.surveymonkey.com/>
* Visual Studio Code Documentation: <https://code.visualstudio.com/docs>