CS 25-333 Quantum Computing for K-12 using Blockly

[Preliminary Design Report]

Prepared for

Thang Dinh - VCU

By

Robert Duncan

Santiago Agudelo

Joshua Pomeroy

Steven Acosta

Under the supervision of

Thang Dinh

Date 12/5/2024

Executive Summary

The object of this Capstone project is to increase K-12 interest in programing by building upon and upgrading a last year's team Capstone project that worked with VCU to use quantum computing to create a Blockly Tic-Tac-Toe game. This game would help kids K-12 to get introduced to the basics of programming with a fun game for the younger students and an easy-to-understand Blockly program to look at for the older students that want to get into the mechanics of programming. QUBO or Quadratic Unconstrained Binary Optimization, is a type of optimization problem used in maximizing and minimizing quadratic objective function without any constraints or decision variables. Our group has been working with our advisor to first understand QUBO and the current iteration of the Tic-Tac-Toe game to be able to upgrade it with new features like saving the game to the ultimate goal creating two new games of Connect 4 and mancala that have a variety of different features including save states, difficulty levels, and game switching to show what Blockly can do and increase our own familiarity with the subject. For the first semester of this project, we have fully updated and finished the Tic-Tac-Toe game with the aforementioned features and a Connect 4 game with many of the same features as the Tic-Tac-Toe game that is nearing completion. We have been able to draw data from the coding we have been doing for the game as well as the setbacks and breakthroughs we have had implementing the games and the best balance to further our project by focusing on engagement, scalability, and educational value when working with the project constraints. We have been keeping track of the project deadlines and goals through a timeline that either shows when we have completed a task or the goal to complete that task by.

Table of Contents

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)

- The goal of this project is to develop a K-12 educational game using quantum computing (QUBO) through Blockly to add to a website that hosts educational quantum computing games

B.2 Design Objectives

- The design will produce one (1) completely functional game of Connect 4 with quantum computing aspects using QUBO

B.3 Design Specifications and Constraints

- The solution should have a simple user interface and visual output so that users of any age can play along and develop their own quantum computing skills

B.4 Codes and Standards

1. **IEEE Standard 1730-2010: Standard for Distributed Simulation Engineering and Execution Process**
    - Relevant for integrating quantum computing simulations and ensuring proper engineering and execution of distributed simulations.
2. **ISO/IEC 25010:2011 - Systems and Software Engineering: Systems and Software Quality Requirements and Evaluation (SQuaRE)**
    - Ensures software meets quality attributes like reliability, usability, and maintainability.
3. **W3C Standard for HTML and CSS (HTML 5.2 and CSS 2.1)**
    - Ensures proper rendering, accessibility, and usability across web browsers and devices.
4. **ISO/IEC 27001:2013 - Information Security Management**
    - Ensures secure handling of information, especially when storing game progress or handling user data.
5. **ACM Code of Ethics and Professional Conduct**
    - Ensures ethical development of the game, protecting user privacy and ensuring age-appropriate content.
6. **ASTM F963 - Standard Consumer Safety Specification for Toy Safety**

    - Ensures digital games for children are safe and age-appropriate.
7. **IEEE 1540-2001: Software Life Cycle Processes - Risk Management**
    - Ensures proper risk management processes during software development, especially during upgrades.
8. **NIST SP 800-63B: Digital Identity Guidelines**
    - Ensures secure authentication and management of user identities.
9. **COPPA (Children's Online Privacy Protection Act) Compliance**
    - Ensures the protection of children's privacy online by obtaining parental consent and implementing privacy protections.

10. **WCAG 2.1 - Web Content Accessibility Guidelines**

    - Ensures web content is accessible to users with disabilities (e.g., visual, auditory, motor, or cognitive impairments).

11. **NIST SP 500-213 - Framework for Cloud Usability**

    - Ensures cloud services supporting the game are usable and efficient.

12. **ISO/IEC 14977:1996 - Information Technology: Syntactic Metalanguage**

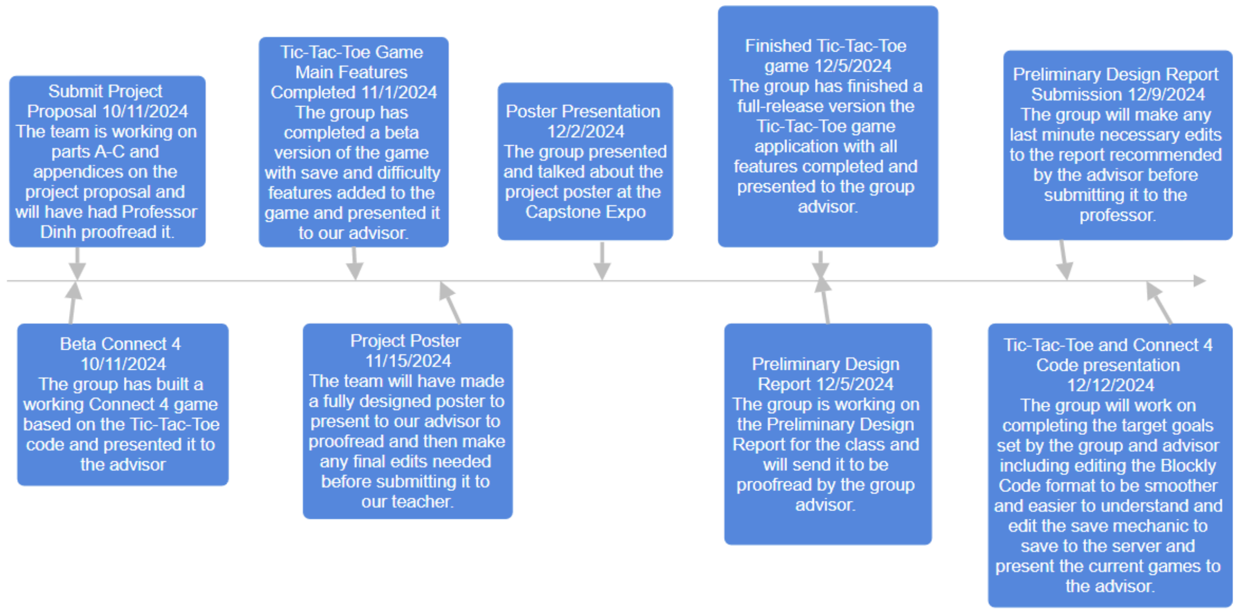
    - Ensures standard syntax representations in the Blockly programming interface.
13. **ISO 9241-11:2018 - Ergonomics of Human-System Interaction**
    - Ensures the user interface is intuitive and ergonomically suitable for K-12 students.

Section C. Scope of Work

C.1 Deliverables

- Website on the internet featuring tic-tac-toe, connect 4, and mancala
- A workspace on the website where Blockly code can be created and used as instructions for the computer-controlled player in the games
- A connection to a quantum computing server within the website, where the computer's moves will be calculated using the server
- Ability to change the difficulty level: make the game engine written in Blockly compete against another predefined game engine.
- Ability to visualize the process of finding the optimal move on the quantum computer
- Ability to save, load, share, and switch between games on the website
- Team contract
- Project Proposal
- Preliminary design report
- Fall poster and presentation

C.2 Milestones



Submit Project Proposal 10/11/2024
The team is working on parts A-C and appendices on the project proposal and will have had Professor Dinh proofread it.

Beta Connect 4 10/11/2024
The group has built a working Connect 4 game based on the Tic-Tac-Toe code and presented it to the advisor

Tic-Tac-Toe Game Main Features Completed 11/1/2024
The group has completed a beta version of the game with save and difficulty features added to the game and presented it to our advisor.

Project Poster 11/15/2024
The team will have made a fully designed poster to present to our advisor to proofread and then make any final edits needed before submitting it to our teacher.

Poster Presentation 12/2/2024
The group presented and talked about the project poster at the Capstone Expo

Finished Tic-Tac-Toe game 12/5/2024
The group has finished a full-release version the Tic-Tac-Toe game application with all features completed and presented to the group advisor.

Preliminary Design Report 12/5/2024
The group is working on the Preliminary Design Report for the class and will send it to be proofread by the group advisor.

Preliminary Design Report Submission 12/9/2024
The group will make any last minute necessary edits to the report recommended by the advisor before submitting it to the professor.

Tic-Tac-Toe and Connect 4 Code presentation 12/12/2024
The group will work on completing the target goals set by the group and advisor including editing the Blockly Code format to be smoother and easier to understand and edit the save mechanic to save to the server and present the current games to the advisor.

C.3 Resources

- 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

Section D: Concept Generation

We considered three main design concepts to create interactive coding games for K-12 students using Blockly and quantum programming:

1. **Concept 1: Standalone Coding Games**
    - **Description:** Each game is designed as a standalone application focused on teaching specific coding concepts (loops, conditionals, etc.) through Blockly integration. The games are modular and can be accessed individually.
    - **Pros:**
        - Easy to develop and test each module independently.
        - Students can focus on specific coding concepts without distractions.
    - **Cons:**
        - Lack of connectivity between games might hinder comprehensive learning.
        - High maintenance due to multiple standalone applications.
2. **Concept 2: Integrated Game Suite with Progressive Difficulty**
    - **Description:** A centralized platform with multiple games where students advance through levels (easy, medium, hard) and are introduced to progressively complex programming challenges.
    - **Pros:**
        - Encourages continuous engagement and skill building.
        - Simplified user experience with a unified platform.
    - **Cons:**
        - Requires significant effort to integrate all games seamlessly.
        - May require more computational resources.
3. **Concept 3: Quantum vs. Classical Coding Games**
    - **Description:** Games that highlight differences between classical and quantum programming through interactive comparisons (e.g., CPU vs. Quantum CPU games).
    - **Pros:**
        - Offers unique value by introducing cutting-edge quantum programming concepts.

        - Encourages curiosity about advanced computational topics.
    - **Cons:**
        - May be too advanced for younger students without strong foundational coding skills.
        - Higher complexity in implementation.

Section E. Concept Evaluation and Selection

	Weight:	Design Concept 1	Design Concept 2	Design Concept 3
Educational Scale:	30%	7	9	8
Engagement Potential:	25%	6	9	8
Scalability:	20%	5	8	7
Technical Complexity:	15%	8	7	6
Resource Requirements:	10%	9	6	5
Total Score:	100%	7.1	8.4	7.3

Selected Concept: Integrated Game Suite with Progressive Difficulty

This concept strikes the best balance between engagement, scalability, and educational value while remaining achievable within project constraints.

Section F. Design Methodology

We received the base Tic-Tac-Toe game and repeatedly added new features as needed from the client. Our design originally consisted of only human vs quantum CPU gameplay and we expanded it to allow the user to set up games between the users choice of a classical CPU, quantum CPU and human players. Along with the addition of CPU players we also added difficulties to the classical CPU that the user can unlock as they progress through a best-of-3 game against the CPU. Along with new gameplay features, we also updated the user interface and added new features to the Blockly interface to increase the accessibility to users of all ages.

We verified these additions though weekly sessions with our client where we would share and demonstrate the working builds we had and would receive feedback to work upon.

F.1 Computational Methods

- **Blockly Integration:** The Blockly framework is used to enable drag-and-drop coding. A custom interpreter translates Blockly commands into executable game logic.

- **CPU Algorithms:** Randomization is implemented for easy and medium difficulties, while the minimax algorithm powers hard mode for optimal CPU behavior.

F.2 Experimental Methods

- **User Testing:** Games will be tested with a sample group of students and educators to assess usability, educational impact, and engagement.
- **Iterative Refinement:** Feedback from testing will guide improvements in game mechanics, visuals, and learning outcomes.

F.3 Architecture/High-level Design

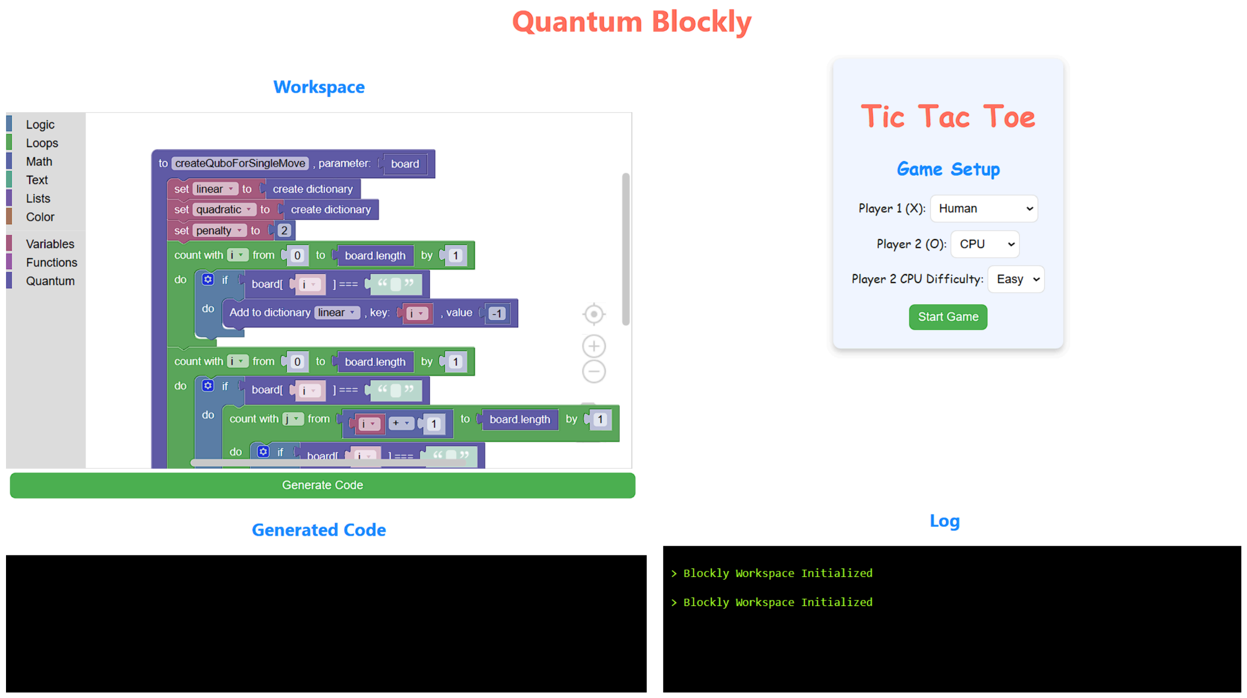
1. **Presentation Layer:** A web-based UI for students to interact with games, built using HTML, CSS, and Javascript.
2. **Logic Layer:** Handles game mechanics, difficulty progression, and Blockly-to-code translation.
3. **Data layer:** Stores user progress, game states, and settings using local storage.

F.4 Validation Procedure

In addition to our verification process, we will present our final build to our client early-April and will give a definitive demonstration of the working product by allowing the client to play with the product to test any cases and the team will also run through the final build to show functionality. Feedback will be received through observation notes while the client plays with the product and then the client will also be interviewed afterwards to collect their thoughts from their experience.

Section G. Results and Design Details

G.1 Modeling Results



```
to createQuboForSingleMove , parameter: board
  set linear to create dictionary
  set quadratic to create dictionary
  set penalty to 2
  count with i from 0 to board.length by 1
  do  if board[ i ] === " "
      do  Add to dictionary linear , key: i , value -1
  count with i from 0 to board.length by 1
  do  if board[ i ] === " "
      do  count with j from i + 1 to board.length by 1
          do  if board[ i ] === " "
```

Player 1 (X): Human

Player 2 (O): CPU

Player 2 CPU Difficulty: Easy

```
> Blockly Workspace Initialized
> Blockly Workspace Initialized
```

G.2 Prototyping and Testing Results (example subsection)

- **Front-end Prototypes:** Initial UI designs for game menus, Blockly coding areas, and game screens have been storyboarded and are under development.
- **Back-end Prototypes:** The logic for difficulty levels, AI algorithms, and game state management has been implemented.

G.3. Final Design Details/Specifications (example subsection)

- **Game Suite:** A centralized platform where students can save and switch between games.
- **Quantum vs. Classical Mode:** Features both a classical CPU and a quantum CPU simulation for side-by-side comparison.
- **Win Streak Mode:** Tracks progress and dynamically adjusts difficulty.

Note that while the design constraints and specifications may have provided minimum or maximum values, or ranges or values, that the design needed to meet, the final design specifications should be listed here showing that the required design values were met. A list of final design details can also be included to demonstrate fulfillment of the design objectives.

Section H. Societal Impacts of Design

H.1 Public Health, Safety, and Welfare

- Encourages digital literacy and computational thinking, crucial for 21st-century education.
- Accessible interface promotes inclusivity for diverse learners.

H.2 Societal Impacts

- Makes programming and quantum computing accessible to younger students, potentially inspiring future technologists.
- Provides educators with engaging tools to teach complex subjects.

H.3 Economic Impacts

- Prepares students for STEM careers, contributing to a skilled workforce.
- Reduces barriers to entry for learning advanced topics, democratizing education.

H.4 Ethical Considerations

- Ensures inclusivity by designing for accessibility (e.g., visual impairments).
- Protects user data through secure local storage practices.

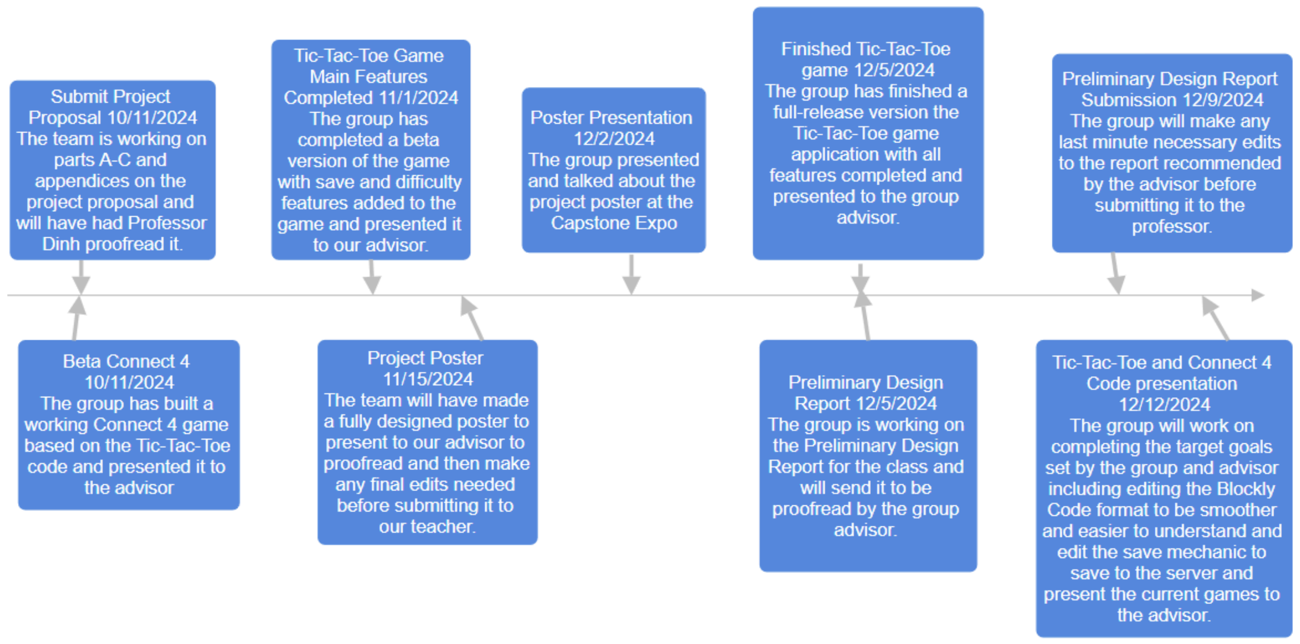
Section I. Cost Analysis

So far, we have not needed to purchase anything to support the development of our application. We have used free, open-source tools for creating and refining our code and did not require any licensed software. In the spring semester, we plan to deploy our application to a real webpage, and this means that we will have to determine the costs of hosting and upkeep that are associated with web pages on the internet.

Section J. Conclusions and Recommendations

We arrived at our final design in three main ways: through testing of the computer opponent, through refining the block-based programming interface, and by improving the user interface of the game. Testing of the computer opponent was done by ensuring that the computer would be reasonably beatable at all levels, if the correct strategy was used. The computer opponent was also tested heavily against different versions of the quantum block code, to ensure that different approaches to programming the quantum server were all valid options for the user. Refining the block-based programming interface was accomplished by adding a zoom functionality to the canvas and a local browser save feature to easily save, load, and condense created blocks of code. The user interface of the tic-tac-toe and new connect four games was updated and went through several revisions before a final version was agreed upon. The interface features smooth edges on the game boxes and an updated font and color scheme. It also features updated menus and better navigation between choosing game options and running the game. These three main factors came together to help us meet our final design objective for this semester: a functional prototype featuring two playable games, multiple game modes, and an improved Blockly programming interface.

Appendix 1: Project Timeline



Submit Project Proposal 10/11/2024
The team is working on parts A-C and appendices on the project proposal and will have had Professor Dinh proofread it.

Beta Connect 4 10/11/2024
The group has built a working Connect 4 game based on the Tic-Tac-Toe code and presented it to the advisor

Tic-Tac-Toe Game Main Features Completed 11/1/2024
The group has completed a beta version of the game with save and difficulty features added to the game and presented it to our advisor.

Project Poster 11/15/2024
The team will have made a fully designed poster to present to our advisor to proofread and then make any final edits needed before submitting it to our teacher.

Poster Presentation 12/2/2024
The group presented and talked about the project poster at the Capstone Expo

Finished Tic-Tac-Toe game 12/5/2024
The group has finished a full-release version the Tic-Tac-Toe game application with all features completed and presented to the group advisor.

Preliminary Design Report 12/5/2024
The group is working on the Preliminary Design Report for the class and will send it to be proofread by the group advisor.

Preliminary Design Report Submission 12/9/2024
The group will make any last minute necessary edits to the report recommended by the advisor before submitting it to the professor.

Tic-Tac-Toe and Connect 4 Code presentation 12/12/2024
The group will work on completing the target goals set by the group and advisor including editing the Blockly Code format to be smoother and easier to understand and edit the save mechanic to save to the server and present the current games to the advisor.

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