Documentation for Vector Analyzer Project

Overview

The Vector Analyzer project is a command-line C# application for performing operations on 3D vectors. It provides users with functionalities to create, store, and manipulate vectors, along with performing mathematical operations. The system is modular and designed with clear separation between data representation, storage management, and mathematical computations.

Getting Started

1. Prerequisites

- Visual Studio 2012 or later installed.
- .NET Framework 4.0 or higher.

2. Running the Application

1. Clone the repository or copy the source code.
2. Open Visual Studio 2012 and create a new Console Application project.
3. Paste the source code of the Vector Analyzer program into the Program.cs file of the project.
4. Ensure there are no errors in the code. If there are, resolve them as prompted by the IDE.
5. Run the application by pressing **F5** or selecting "Start Debugging" from the Debug menu.
6. Interact with the application through the command-line interface that appears.

Key Components

1. Vector Class

- Represents a 3D vector with components `x`, `y`, and `z`.
- **Constructor:**

  ```
  Vector(string name, double x, double y, double z)
  ```

- **Methods:**
  - `double GetMagnitude()`: Computes the vector's magnitude.
  - `Vector GetUnitVector()`: Returns the unit vector.
  - `static Vector Parse(string input)`: Parses a string (e.g., "v = 3i + 4j - 2k") to create a vector.
  - `static double GetAngleBetweenVectors(Vector v1, Vector v2)`: Calculates the angle between two vectors in degrees.

2. Store Class

- Manages a collection of vectors.
- **Constructor:**

```
Store(uint size)
```

- **Methods:**
  - `bool AddVector(Vector v)`: Adds a new vector to the store.
  - `bool RemoveVector(string name)`: Removes a vector by its name.
  - `void ListVectors()`: Displays all stored vectors.
  - `Vector GetVector(string name)`: Retrieves a vector by its name.

3. VectorOperations Class

- Contains static methods for performing mathematical operations.
- **Methods:**
  - `static Vector AddVectors(string name, Vector v1, Vector v2)`: Adds two vectors.
  - `static Vector SubtractVectors(string name, Vector v1, Vector v2)`: Subtracts one vector from another.
  - `static Vector ScalarMultiplication(Vector v, double scalar)`: Multiplies a vector by a scalar.
  - `static double DotProduct(Vector v1, Vector v2)`: Calculates the dot product of two vectors.
  - `static Vector CrossProduct(Vector v1, Vector v2)`: Computes the cross product of two vectors.
  - `static double Projection(Vector ofVector, Vector overVector)`: Finds the scalar projection of one vector over another.

4. VectorAnalyzer Class

- Provides the main menu and coordinates user interactions.
- **Menu Options:**
  - Vector Store (Add, Remove, Modify, List vectors)
  - Vector Addition, Subtraction
  - Scalar Multiplication
  - Dot Product, Cross Product
  - Angle Between Vectors
  - Projection

How to Use

1. Creating a Vector

- Input format: `v = 2i + 3j + 4k`
- Example:

```
Input a vector along with its name (e.g., v = 2i + 3j):
v = 3i - 4j + 5k
```

2. Performing Operations

- Navigate through the menu to:
  - Add vectors to the store.
  - Compute dot/cross products.
  - Calculate unit vectors, projections, or angles.

3. Example Session

```
-------------------------------------------------------
                        MAIN MENU
-------------------------------------------------------
0. Exit
1. Vectors Store (To add, remove, modify, see vectors)
2. Addition of Vectors
3. Subtraction of Vectors
4. Scalar Product
5. Dot Product
6. Cross Product
7. Projection
8. Redisplay Menu

8 to Redisplay Menu
Select an Option: 1
Welcome to Store

-----------------------------------
          VECTOR STORE MENU
-----------------------------------
0. Exit form Store
1. Add a new vector
2. List Vectors
3. Remove a vector
4. Modify a vector
5. Explore a Vector
6. Redisplay Store Menu
7. Display Unit Vector
8. Calculate Angle Between Two Vectors

6 to Redisplay Menu
Select an Option(Store):
```

Error Handling

- Invalid input formats are rejected with an appropriate error message.
- Operations involving zero vectors are disallowed (e.g., angle calculation).
- Store capacity is capped at 20 vectors.

Future Enhancements

- Add file input/output for vector storage.
- Extend the program to handle n-dimensional vectors.
- Develop a graphical user interface (GUI).

Contributors

- Siraj Shabbir
- Muhammad Farhan Ali