# Assignment Description

You are working on the newest racing game.  Your part is to calculate the speed the car is moving based on acceleration and time.  The game determines the acceleration based on how hard the user is pressing the analog trigger button or racing pedal. While the rest of the game is being written, you will write the functions that meet the specifications below.

# GitHub URL (optional)

[https://github.com/wesleyhixon/Programming-Assignments/tree/8f32361c8c469e685a92fa61fcf8a5000bb8ca16/M05%20Programming%20Assignment%201](https://github.com/wesleyhixon/Programming-Assignments/tree/8f32361c8c469e685a92fa61fcf8a5000bb8ca16/M05 Programming Assignment 1)

# Readme Documentation

Input Information: An input force 1-10 and a total acceleration time

Output Information: The acceleration in m/s^2 and a table showing acceleration over the given time.

# Flowchart Screen Shots (optional)

Screen shot(s) here

# UML and Use Case Diagrams (optional)

Screen shot(s) here

# Source Code of All files (.h, .cpp)

#include *<iostream>*

#include *<iomanip>*

**using** **namespace** **std**;

*/\**

*Program Name: Acceleration*

*Author: Wesley Hixon*

*Date Last Updated: 07/02/2024*

*Purpose: Given a button force and accleration time, output a table of acceleration over time.*

*\*/*

*// This function calculates speed given acceleration and time*

double speedDueToAcceleration(double acceleration, int time){

double s;

s = acceleration \* time;

**return** s;

}

*// This function calculates the acceleration from force input*

double accelerationFromButton(double buttonForce){

double a;

a = buttonForce \* 0.463;

**return** a;

}

int main()

{

bool valid = false;

int forceInput, totalAccelerationTime;

double speed, acceleration;

*// Getting user input for button force and validating it*

**while**(!valid){

cout << "Enter the button force between 0 and 10: ";

cin >> forceInput;

*// in case of invalid number*

**if**(forceInput > 10 || forceInput < 0){

cout << "The button force needs to be between 0 and 10. Please try again. **\n**";

**continue**;

}

*// in case of input failure*

**else** **if** (!cin){

cout << "You entered something that is not a number. Please try again. **\n**";

cin.clear();

cin.ignore(10000, '\n');

**continue**;

}

**else**{

valid = true;

}

}

*// resetting valid for the next input*

valid = false;

*// getting user input for acceleration time and validating it*

**while**(!valid){

cout << "Enter the total acceleration time in seconds: ";

cin >> totalAccelerationTime;

*// in case of invalid number*

**if**(totalAccelerationTime < 0){

cout << "The acceleration time must be greater than 0. Please try again. **\n**";

**continue**;

}

*// in case of input failure*

**else** **if**(!cin){

cout << "You entered something that is not a number. Please try again. **\n**";

cin.clear();

cin.ignore(10000, '\n');

**continue**;

}

**else**{

valid = true;

}

}

*// calculate acceleration from the force input*

acceleration = accelerationFromButton(forceInput);

*//output table header*

cout << "Acceleration is " << setprecision(3) << acceleration << " m/s^2" << endl;

cout << left << setw(15) << "Time in s" << setw(15) << "Speed in m/s" << endl;

cout << setw(30) << setfill('\*') << "\*" << endl;

cout << setfill(' ') << fixed << showpoint;

int time = 0;

*// This is the loop which outputs the acceleration table*

*// It continues running until the total acceleration time is exceeded*

**while**(totalAccelerationTime - time >= 0){

*// Calculating the speed and outputting the next line of table*

speed = speedDueToAcceleration(acceleration, time);

cout << setw(15) << time << setw(15) << speed << endl;

*// This handles the case when totalAccelerationTime is not divisible by 3*

*// When totalAccelerationTime - time is < 3 AND time does not equal totalAccelerationTime*

*// Then, add the remaining difference to time*

**if** (totalAccelerationTime - time < 3 && totalAccelerationTime != time){

time += (totalAccelerationTime - time);

}

*// Otherwise, simply increment time by 3*

**else**{

time += 3;

}

}

**return** 0;

}

# Three Use Case Screen Shots

The can all be in a single screen shot





