Cruise Control
Software Development Document
V.0.6

Team SCJB:

Sunmin Lee

Cassie Ball

Julia Nelson

Seung (Ben) Eom

"I pledge my honor that I have abided by the Stevens Honor System."

Table of Contents

1. Executive Summary
2. Introduction
    a. Customer need
    b. Major features
    c. Reliability
    d. Performance
3. Requirements
    a. Functional Requirements
        i. Inputs
        ii. Outputs
    b. Systems Requirements
    c. Non-Functional Requirements
        i. Performance
        ii. Reliability
        iii. Availability
        iv. Maintainability
        v. Security
4. Requirements Analysis Model
    a. UML Use Cases & Diagrams
        i. Use Case 1
        ii. Use Case 2
        iii. Use Case 3
    b. UML Class-Based Modeling
    c. UML CRC Model Index Card
    d. UML Activity Diagram
    e. UML Sequence Diagram
    f. UML State Diagram

1. Executive Summary

This project is to develop a cruise control that is used to automatically maintain a car’s speed without the user having their foot on the gas pedal. Cruise control is a feature offered in the majority of modern cars to provide user convenience specifically for long drives on highways. Some major features that are needed to make the cruise control work include turning the program on and off, a dashboard display of the speed of the car, buttons to increase and decrease the speed as necessary, and so on. Our goal is to successfully implement these major features with no chance of failure in order to increase the reliability of our cruise control software and enhance the overall performance of the cruise control system.

2. Introduction

High-Level Customer Need

Cruise control is a highly sought after feature in vehicles that electronically interacts with the engine's management system to maintain its speed. Originally invented to save gas by reducing gas-wasting surges on the accelerator, cruise control provides an opportunity for the driver to rest their foot and safely adjust their positioning, while also helping to prevent speeding tickets on long roads and highways.

Major Features

Our implementation of cruise control needs a number of standard features in order to work properly. The first major feature we need is to take input that will turn cruise control on and off. In addition, cruise control needs to be able to shut off when the user presses the brake and, when the car turns off, the entire program should stop running. We need another input that sets the speed that the cruise control will maintain. To see the speed that we are maintaining we will need a dashboard display of a speedometer that shows the current speed. Input to increase and decrease the set speed of the cruise control will also be implemented so the driver does not have to turn off cruise control and try to get the desired speed by pressing the gas. There will be a logging system that writes data to an excel spreadsheet as soon as cruise control is turned on. It will track any changes in the program such as when it turns on, being set, the current speed, and when it is turned off or deactivates. This will provide a better understanding of how the cruise control program is running and if there is a problem, it will be easier to understand where and why it occurred.

High-Level Reliability

As the reliability of our implementation of cruise control is directly related to the safety of the passengers, it is crucial to develop highly reliable software to give the passengers a sense of security. Ultimately, our goal will be for our implementation to have a success rate of 99.99% for hardware and 99.999% for software of the cruise control. To achieve this goal, we have to make sure that all of the major features mentioned above, such as turning the cruise control on and off, and setting the speed the passenger wants to maintain the vehicle at, should work 99.99% of the time. With our agile development process, we will successfully implement all the features with a low probability of failure and test the reliability of our implementations via series of tests and experiments, to make sure that we have reached our goals.

High-Level Performance

The cruise control system will have to have optimal performance while the car is in motion. Optimal performance means that the vehicle will reach the cruise control speed that is set by the driver as soon as possible. The cruise control will also have to maintain the speed that the driver decided to set by the cruise control button until the driver turns off the engine or turns off the cruise control option. It is important that once the cruise control is turned on, and the speed has been set, that the vehicle does not dip above nor below the speed. A fluctuation of the speed of the car is an indicator of a low-level performance, which should be avoided and be accounted for when implementing the cruise control system in the vehicle.

3. Requirements

Functional Requirements

Input (Requires user input to activate)

FR-I-1. On an activate input, cruise control shall turn on and display that cruise control has been turned on, and set to the current speed that the driver was already driving on.

FR-I-2. On a cancel input, cruise control shall turn off completely and the cruise control display icon will turn off.

FR-I-3. On a set speed input, cruise control shall maintain the speed of the car at the current speed shown on the speedometer.

FR-I-4. On an increase speed input, cruise control shall increase the set speed by 1 mph.

FR-I-5. On a decrease speed input, cruise control shall decrease the set speed by 1 mph.

FR-I-6. On an acceleration input, the car shall speed up. When the accelerator is released, the car shall return to its cruise control set speed. If the driver wishes so, the new speed can be set again by a set speed input.

FR-I-7. On a brake input, cruise control shall turn off the set speed but the cruise control system is still activated, and will wait for a new set speed.

FR-I-8. When the car turns off, cruise control shall turn off completely.

Output (Functions that output without user)

FR-O-1. Logging of car data shall begin within 1 second of the car turning on.

FR-O-2. When the vehicle shuts down, cruise control hardware shall perform a graceful automatic shutdown after 1 minute, as to not consume the car's battery electricity.

System Requirements

SR-1. The cruise control system shall have a sensor to detect the speed of the car.

SR-2. The cruise control system shall log all changes of speed and operations during usage.

SR-3. The cruise control shall perform an automatic shutdown after one minute to preserve battery if cruise control has been turned on, but a speed has not been set.

Non-Functional Requirements

Performance

NF-P-1. The cruise control system shall display that it is turned on within 100ms of an ON input from the driver.

NF-P-2. The cruise control system shall turn off within 100ms of a cancel input.

NF-P-3. The cruise control system shall deactivate the set speed within 100ms of a brake input.

Reliability

NF-R-1. The cruise control system shall be available to use 99.99% of the time.

Availability

NF-A-1. The cruise control system shall be ready to use without delay 99.99% of the time.

NF-A-2. The cruise control system shall be ready to turn on within 2 seconds after the car is turned on and the engine starts running.

NF-A-3. The cruise control system shall only be able to maintain a speed once the car is driving at a minimum speed of 25 mph.

Maintainability

NF-M-1. The cruise control shall be able to maintain itself until the car becomes unable to move. If maintenance is needed, the cruise control unit shall be able to signal to the primary actor/driver whenever the system is unavailable.

NF-M-2. The technician shall be able to view the logged data including all state changes of the cruise control system from turning on, current speed, and turning off.

Security

NF-S-1. The cruise control unit shall not be connected to the internet, so it will not be vulnerable to cyber attacks. The only vulnerability it has is if there is a physical harm against the unit. If the cruise control system were to fall to an physical attack that caused any controls or sensors to malfunction outside optimal operation, the cruise control system would alert the driver and deny usage.

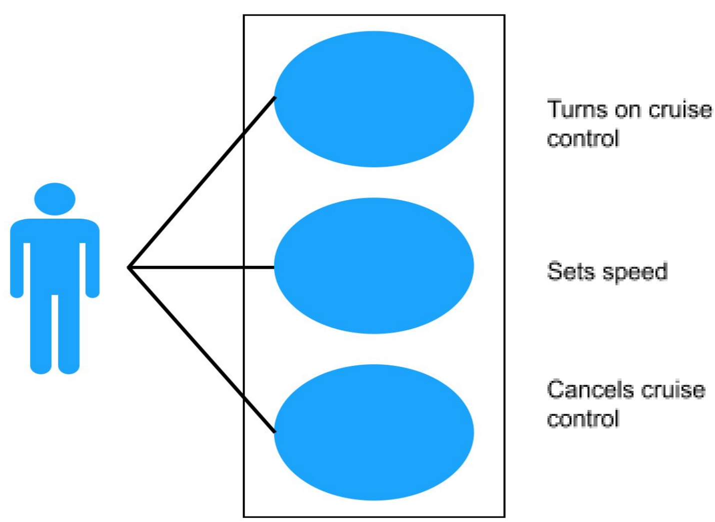
4. Requirements Analysis Modeling

UML Use Cases

Use case 1: Usage of Cruise Control on a highway
Primary actor: The driver
Goal in context: Use cruise control to drive on a highway.
Precondition: The car is in good condition, and the driver is trying to use cruise control in an appropriate condition (on a highway, with a speed of over 25 mph)
Trigger: Driver decides to use cruise control on a highway

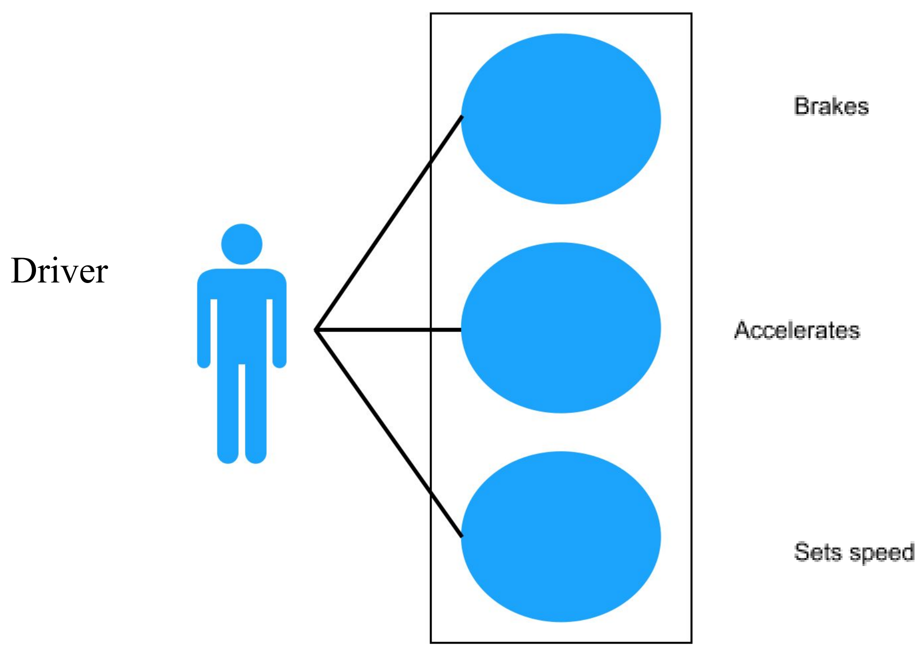
1. While on the highway, the driver powers on cruise control through the user interface.
2. Cruise control gives visual feedback to show that it has been activated.
3. The driver sets it to the speed that he wants, and visual feedback shows the current speed.
4. Cruise control requests values from sensors. Sensors provide approval to set cruise control at current speed, and cruise control requests the engine management system to set the speed at current position for desired speed.
5. The Engine Management System is set to the indicated speed, and is visually shown to the driver. The driver can adjust according to changes of surrounding environment, and the sensors shall provide feedback, resulting in changes of the speed of EMS too.
6. The driver increases and decreases set speed and the change is shown on the speedometer.
7. The driver is done using cruise control and hits cancel which turns the cruise control system off completely.

Driver



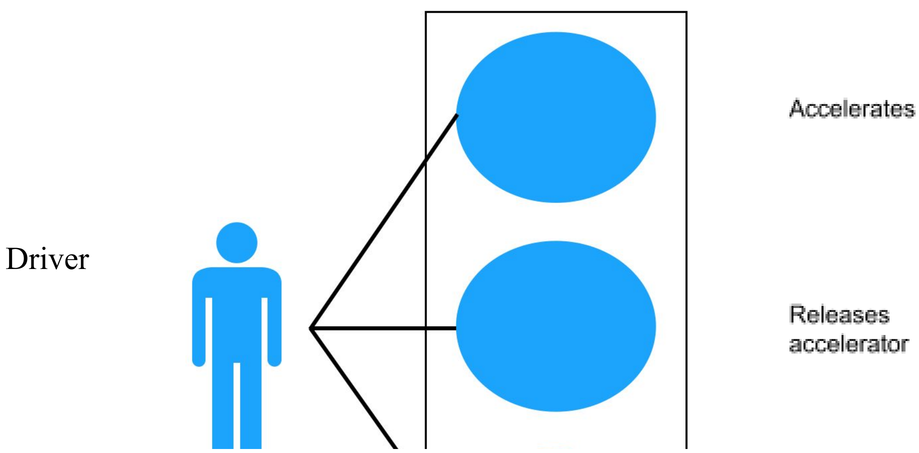
Use case 2: Usage of brake with cruise control engaged.
Primary actor: The driver.
Goal in context: Cruise control shall disengage when brake is applied.
Precondition: The car is in good condition, and the driver has activated and engaged cruise control in an appropriate condition (on straight road, with a speed of over 25 mph)
Trigger: Driver must brake to avoid potential collision.

1. While using cruise control, the driver sets foot on the brake pedal.
2. Cruise control disengages to a neutral state of activation, no longer controlling the vehicles throttle.
3. Cruise control gives visual feedback to show that it has been disengaged.
4. The driver accelerates again to their desired speed
5. The driver sets the cruise control speed.
6. Cruise control requests values from sensors that provide approval to set cruise control at the current speed. Cruise control requests the engine management system to set the speed at current position for desired speed and then displays visual feedback of the current speed.

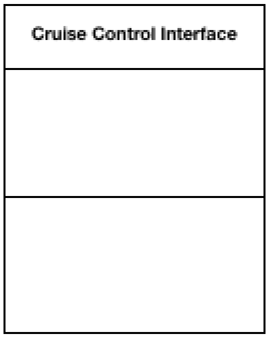


Use case 3: Usage of accelerator with cruise control engaged.
Primary actor: The driver.
Goal in context: Cruise control shall allow acceleration and then re-engage at the originally set speed when the accelerator is released.
Precondition: The car is in good condition, and the driver has activated and engaged cruise control in an appropriate condition (on straight road, with a speed of over 25 mph)
Trigger: Driver wants to use the accelerator to pass another car on the road.

1. While using cruise control, the driver sets foot on the accelerator pedal.
2. Cruise control shall disengage, no longer controlling the vehicle's throttle, and allow acceleration of the vehicle.
3. Cruise control gives visual feedback to show that it has been disengaged.
4. The driver releases the accelerator and the vehicle begins to decelerate. Once the speed of the vehicle begins to decelerate, cruise control shall read the speed log until the previously set speed is reached. Then cruise control shall request the engine management system to maintain the speed at the current position.
5. Cruise control shall then display visual feedback of the current speed to the driver.



UML Class-Based Modeling



UML CRC Model Index Card

Class: Cruise Control On	
Activates the cruise control system Keeps the cruise control on standby, waiting for set speed Keeps the cruise control on standby, waiting for the increase speed Keeps the cruise control on standby, waiting for the decrease speed Will disengage when “cruise control off” is activated	Set speed Increase speed Decrease speed Cruise Control Off

Class: Cruise Control Off	
Deactivates the cruise control system Disables the “cruise control on”	Cruise Control On

Class: Set Speed	
Sets the speed of the cruise control system Sets the new speed to the increased speed that is set by driver Sets the new speed to the decreased speed that is set by driver	Increase speed Decrease speed

Class: Increase Speed	
Increases the desired speed that can be set by the driver	

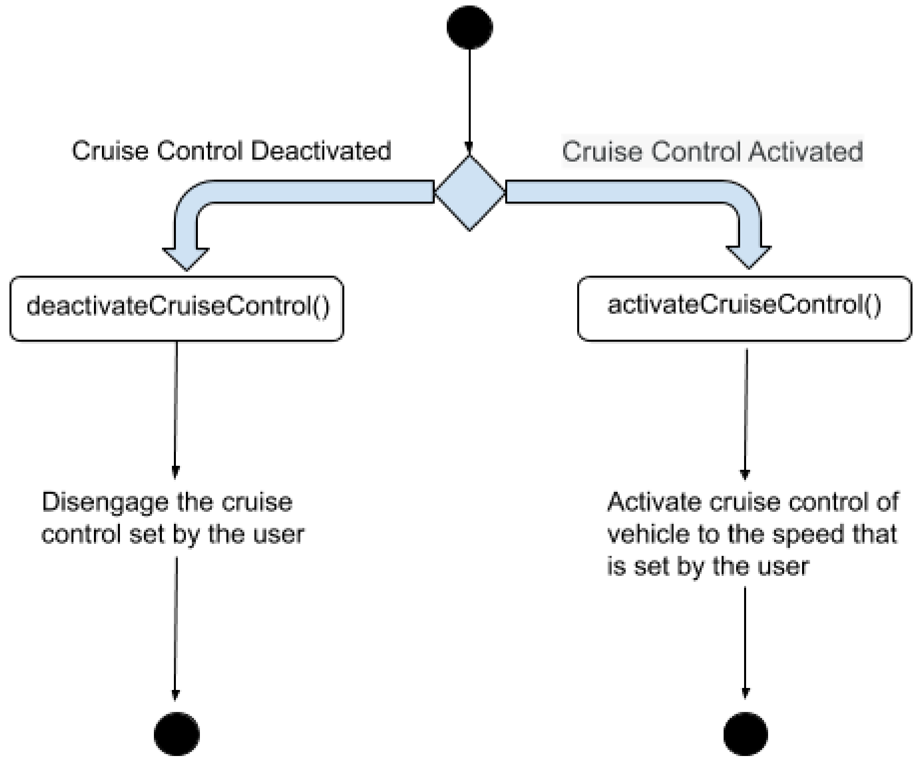
Class: Decrease Speed	
Decreases the desired speed that can be set by the driver	

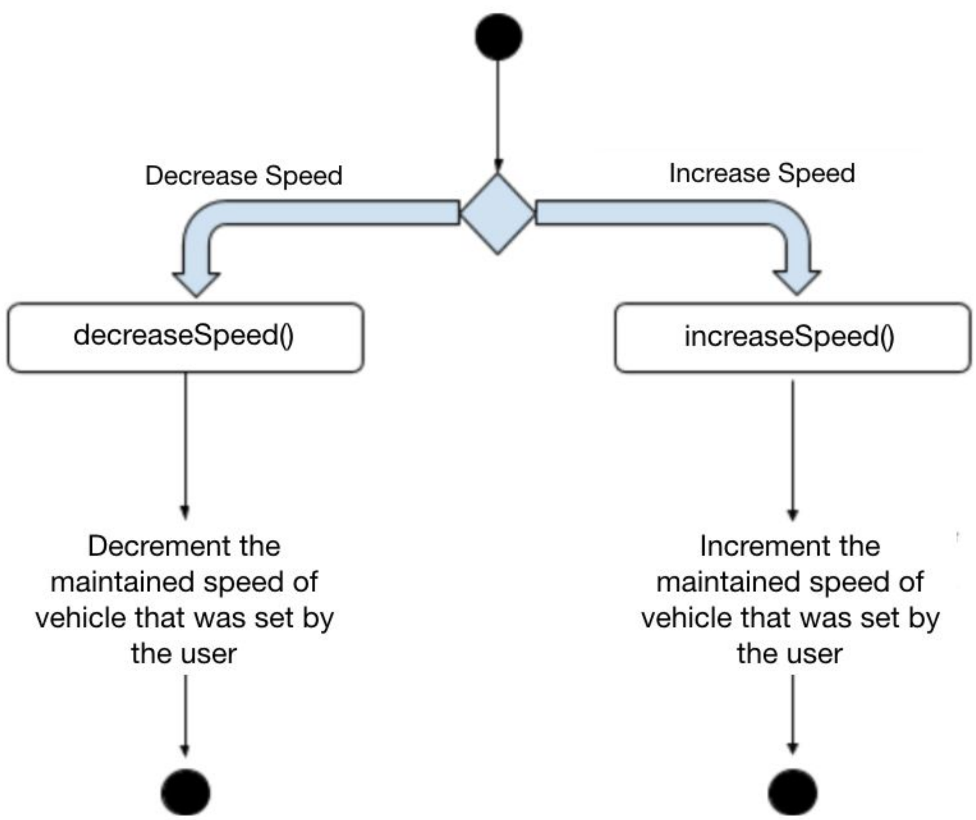
Class: Set Speed	
Sets the speed of the cruise control system Sets the new speed to the increased speed that is set by driver Sets the new speed to the decreased speed that is set by driver	Increase speed Decrease speed

Class: Accelerator Pedal Input	
Increases the speed of the car Speed is increased past the speed that is set by the cruise control system After the acceleration pedal is released, cruise control shall maintain originally set speed upon deceleration.	Set speed

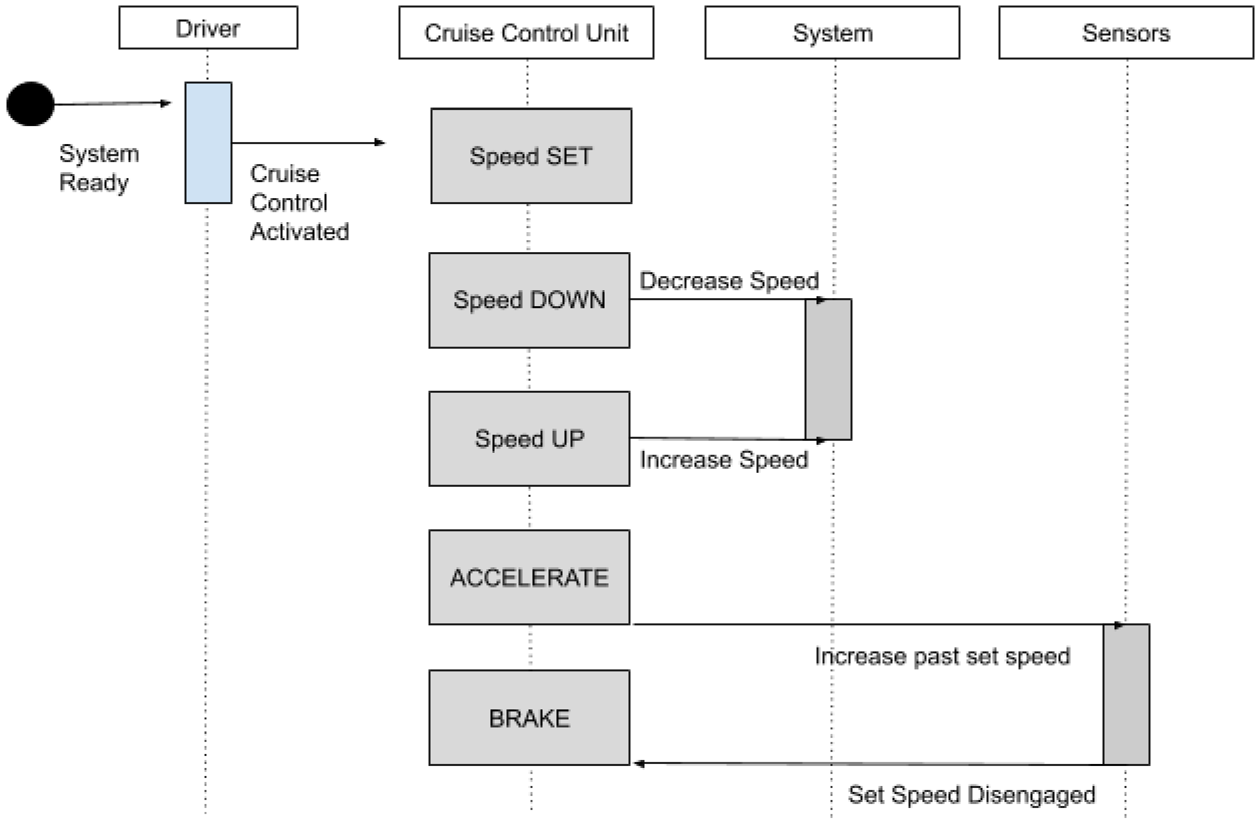
Class: Brake Input	
Vehicle will start to decelerate, but cruise control will still be active Cruise control system will still stay activated The vehicle will not have the set speed anymore	Cruise control on Set speed

UML Activity Diagram





UML Sequence Diagram



UML State Diagram