

Car systems (LM4)

Purpose

This software is expected to allow users to learn more about cars and how do they work. The main idea is to introduce to the user various systems of the car and how they all interlink together in order for the car to operate as one product.

It is designed to be a fun and easy to use software to ensure the users of it are able to effectively learn about cars.

Marketing and competition

It is expected to benefit both new drivers and current drivers who would like to learn more about the various systems in the car. There aren't many competitors for this kind of software in the market at the current point in time, mainly because many still rely on books or other printed media.

It is expected due to the lack of many competing software, our solution is likely to be quite revolutionary in the way it delivers information about cars, which in the past has been viewed to be not that exciting.

SDLC Methodology

We will be implementing a spiral model, made up of the following stages:

- Brainstorm: Brainstorming ideas to see what other features or improvements can be made to the system
- Research: Research potential implementations of this feature that has been proposed
- Programming: If viable, we will then write the code out for this functionality
- Testing: Test out the code thoroughly to ensure it works
- Release: Integrate it into the system and once successful, start working on the next feature

Development

As we intend to deploy on Windows computers only, specifically Windows 8 and above, we have decided to use Visual Studio (VS) as our choice of our development environment for the following reasons:

1. In comparison to Visual Basic, VS allows for more flexibility in designing the user interface of the system
2. Unlike Qt, VS is significantly more user friendly, making it easier for us to work on the code, which would allow us to get more work done faster and better
3. There is an active and large development community for VS, allowing better research to be done
4. It comes with support for Visual C++, which is our choice to do the UI in
5. For students, VS is available for free

We decided against game engines because our group doesn't have anyone who has had past experience in game engine. This would be a steep learning curve that will hinder the amount of features we can implement.

Technology

The following technologies will be used:

1. C++ : Full object oriented programming, allows us to manage the threads on our own, so we have full control

2. Multithreading: We plan to have each main feature of the system in its own thread. In addition, for every simulation, each component of the car is expected to run in their own threads, and to link all the threads up, there should be some form message passing system that has to be implemented.
3. Git: We are using git for version control, as it allows us to track changes easily