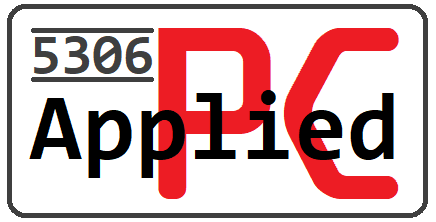
****Assignment 1

Goal Identification

1. **Overview**

Super Tux Kart (STK) is a 3D open-source arcade racing game with a variety of characters, tracks, and modes to play. Everything you need to know about STK can be found on their website: https://supertuxkart.net/Main\_Page. The most recent version is 1.3 (2021-09-28). For this assignment, you will probably wish to access the source code (please see https://supertuxkart.net/Source\_control). For reference, any data I provide is taken from:

stk-code-master.zip (uncompressed: 76.6 MB, 3393 Files in 181 Folders).

I consider STK to be a modest sized software system. It is mature, well developed, highly used, and well respected. For this assignment, we are going to work with STK as it can be compiled for Windows, Linux, and OS-X, it is free, and it has never been used in this course (so no solutions are available for the assignments).

Your hypothetical task (which you won’t perform) is to modify STK by adding a new feature: “Oil slicks” – your car dumps a big patch of oil on the ground behind it that causes other cars to go into a spin and crash.

You are going to develop a set of goals (i.e., questions) to guide the program comprehension that is necessary to perform the hypothetical task (i.e., adding Oil Slicks to STK).

1. **Requirements**

**Groups**: You are to work in your assigned “Code Crew.”

**System**: For this assignment, you are to work with the latest stable version of STK (see above). You may work with any variant you wish (i.e., OS-X, Windows 10, Linux).

**Task**: Develop a detailed, accurate, realistic, and well-defined set of goals to drive the program comprehension that is needed for the successful extension of STK with the Oil-Slick feature. You are not going to modify the code in any way; all you are doing is developing the set of goals.

1. **Deliverable**

Each group will produce a PDF document that details the group member’s experiences. Your document should contain at least:

1. An introduction,
2. A detailed list of goals with explanations for why you selected them,
3. A summary of your procedure and experiences,
4. A conclusion with descriptions of the main things you learned, and
5. A list of references.

Please note that this is a suggestion on how to structure your document; it is not a definitive or required set of section headings. I value and reward students that think independently and can structure their thoughts in a clear and concise manner. Your document should describe your experiences and report your failures as well as your successes. Explain what didn't work, and why, as self reflection assists you to more fully learn the art of program comprehension.

There is no maximum or minimum length requirement, though I would expect your document to be more than a couple of pages in length. You must include a complete bibliography of resources, and materials used or referenced. A list of URIs is not a suitable bibliography; please use **the IEEE citation** and reference style. See:

https://www.ijssst.info/info/IEEE-Citation-StyleGuide.pdf

for details.

Your document should be professional and complete. Features, such as a cover page, table of contents, glossary, and appendices are welcome and encouraged. Please do not use full justification; I prefer left justified (i.e., ragged right) documents because they are easier for me to read. Also, please use a serif (as opposed to sans serif) font, such as Century Schoolbook or Times Roman. Use of slang, excessive use of acronyms and jargon, and use of lists (as opposed to complete sentences) are all to be avoided when possible. “Point-form” text is not suitable for academic work.

1. **Evaluation**

Your submission will be evaluated with respect to the following criteria.

Quality of presentation (25%): Professionalism, grammar, spelling, graphic design, layout, structure/organisation, **diagrams**, and other factors that impact the overall quality of the document. Your document should have suitable high-quality references in IEEE style.

Level of detail (Completeness; 25%): More is generally better. I want to see that you dug deep into the code and did more than a quick examination of a few trivial components.

Correctness and conciseness (Accuracy; 25%): I want to be convinced your goals accurately describe how to start finding information so that you can perform the hypothetical task. Irrelevant material that simply adds length without adding value will result in a lower mark; conciseness is a critical component of accuracy.

**Process and experiences** (25%): I want to know what you did. What tools and techniques did you try? What worked and what didn’t? When you encountered problems, how did you solve them?

**Most importantly, have fun!**