CS 350 Task 1: Project Grounding and Conceptualization

This task establishes a guided introductory framework for the primary elements of the quarter project. In real life, you would somehow have to assemble the equivalent of such a list yourself. It would be far larger, more complex, and messier, and would take much longer to compile. Based on the high-level context being discussed in class at this point, your job is to perform preliminary background research on each element in this outline. At this inception phase in the top-down development process, your answers will necessarily be general. In real life, it would be your responsibility to compile something to the effect of this list based on a subjective breadth and depth of coverage. Nobody would ask to see it, but it would be clear if you had not done it. This list establishes the vocabulary and context for understanding of the problem domain and the initial design thinking for the solution domain to come.

Lecture 8 covers the definitions of the bold words.

1. Briefly discuss what each element means to you so far with respect to your current understanding of the project as introduced in the kickoff "meeting" in lecture. Do not just provide a blanket definition or something factually correct but useless; e.g., dogs bark, or the APU does APU stuff. Your interpretations ultimately may not be relevant to our actual solution later, but they must be arguably within its purview. You must address each element separately as structured here and cite its primary source in the format `[@url]`, where *`url`* is the complete text link to your reference (minus any `http[s]://`). (Do not make it a dynamic embedded link.) Use at least six sources in total; Wikipedia is acceptable, but it serves better as a starting point (a resource) than actual information (a reference). Put the term in bold. Indicate your name at the top of the document and the total word count at the end.
2. Indicate the grammatical *category* as **noun**, **adjective**, **preposition**, **verb**, or **other**.

For each element:

3. Briefly address these aspects from a consistent, coherent, practical, computational perspective. If multiple interpretations are possible, choose the most representative. Include the head words here in italics, as shown in the example below:

   data: what it is; properties that describe its existence. For each, indicate whether it is primarily **static** (unchanging) or **dynamic** (changing) and why
   control: what it can do; actions that describe its capabilities
   behavior: what it actually does or is done with it; appropriate actions to satisfy a goal; user stories or use cases

   The entries should be reasonably consistent; e.g., if control acts on something, then corresponding data is likely.
4. Indicate the *role(s)*: **input**, **processing**, or **output** and why.
5. Indicate the paradigm(s) of design *pattern*: **creational**, **structural**, or **behavioral** and why.
6. Indicate the *concern(s)* of an MVC architecture: **model**, **view**, or **controller** and why.
7. Indicate the expected *difficulty* to manipulate it programmatically: **easy**, **moderate**, or **hard** and why.
8. Indicate the expected *risk* of not being able to do it or doing it wrong: **low**, **moderate**, or **high** and why.
9. Indicate your *confidence* in your answer: **low**, **moderate**, or **high** and why.
10. Describe a plausible (for this course) two or three-dimensional visual *presentation* of the content.

Use this solution as an example. It includes more than you need to do.

1. **landing gear, main**: supports and stops an aircraft while on the ground; retracted in flight

source:	[@en.wikipedia.org/wiki/Landing_gear]
category:	noun
data:	wheels (static because they do not need to rotate in our project); strut (dynamic because it needs to extend and retract)
control:	wheels have braking; strut can extend and retract
behavior:	the pilot commands the gear to retract after takeoff and to extend before landing
role:	output because it is a mechanical device that supports and stops a plane
pattern:	creational because it must be defined; structural because it connects to the airplane
concern:	model because it is something manipulated; view because we need to see the state
difficulty:	moderate because it involves motion between two states
risk:	high risk because it involves code for motion; not having this prevents the gear from functioning
confidence:	low because I know nothing about this subject
presentation:	a down-arrow for extended, up-arrow for retracted

A. Address all the following elements in order:

1. acceleration
2. cab signaling
3. catenary wire
4. coordinates, absolute
5. coordinates, relative
6. coordinates, world
7. crossing gate
8. locomotive, diesel
9. locomotive, diesel-electric
10. locomotive, electric
11. locomotive, multiple unit operation (master/slave)
12. pantograph
13. Positive Train Control
14. push-pull train
15. rail yard
16. roundhouse
17. semaphore
18. sensor
19. signal light
20. speed
21. stock, rolling
22. switch (track)
23. track, main line
24. track, siding
25. track, spur

B. Group the elements from (A) (by name only) into a higher-level organization that makes sense to you. For example, *head, body, tail, legs, teeth, eyes, claws, paws, ears, nose, bark* for a dog could be organized as:

<u>senses</u>
eyes
ears
nose

<u>movement</u>
legs
paws

defense
claws
teeth

communication
bark
tail
paw

and so on…

Some terms may belong to more than one category. Be sure to account for all elements. There is no order within a group.

The real world is very messy with ill-defined boundaries. Not everything here will fall cleanly into the specified partitioning. Use your best judgment. Team collaboration would iron out the inconsistencies to establish a consensus.

Tasks C and D apply to the spreadsheet posted with this task. Do not otherwise modify the existing structure or contents.

C. Indicate the bold designations from 2 through 9 for each element as a lowercase **x** in the corresponding cells.

D. Add three elements related to this element respectively in the Related Elements columns. Limit each to a word or two, and do not clarify with parenthesis or commas. Do not reuse any of the elements from (A).

Deliverables

Submit Parts A and B in a PDF document. Submit Parts C and D in CSV format. Incorrect formats will incur a stiff penalty.

Start early. Do not underestimate the time it will take to do this work. The analysis stage of a project is critical to do as completely and correctly as possible. Nobody is born knowing this stuff, and there are no shortcuts to learning something about it. This course is not about trains; it is about applying software engineering to a concrete example that we can work with. This same process works with any project.

Assessment

This is a software engineering class that is designed so you learn how to analyze problems, design and implement solutions, and verify that you have done so correctly. There is no difference in this respect between establishing and following a process for doing this with code or non-code. Remember, over 80% of software engineering involves non-code, such at this document. Also remember that most errors occur in thinking and understanding, not in execution (or errors in execution stem from the former). Take this process seriously because it is helping you to train your mind to read, understand, plan, execute, verify, and reflect at all levels of software development. It is your responsibility to determine whether you understand this task and to complete it properly. Ask if you are unsure.

Follow our process: read ➜ understand ➜ plan ➜ execute ➜ verify ➜ reflect