UF UNIVERSITY of FLORIDA
Herbert Wertheim College of Engineering

Computer Engineering
CEN3907C

M0: General Project Requirements

Overview

This document outlines expectations for students' projects and presentations, which will be completed on teams.

Project Requirements

Project development and submission must follow specific guidelines depending on the nature of the project. Deviation from these criteria will result in a grade of ***zero***. Projects must at a minimum demonstrate mastery of Computer Engineering topics. Projects may focus on software or hardware but must demonstrate understanding of software and hardware. If off the shelf components are used (e.g., libraries, Raspberry Pi, Arduino), ***the project should retarget to compensate*** – i.e., these tools allow projects to go further, not to require less effort.

Project Type

Project requirement will vary in part depending on the type of project. Where there is overlap, teams should target the primary objective of the project when considering requirements.

Tools

1) Intended for interaction directly with the end-user
2) Primary purpose is to accomplish one or more tasks (vs entertainment)
3) Build specifications will vary depending on platform
4) Has target functionality and aesthetic theme

Entertainment

1) Intended for interaction directly with the end-user
2) Primary purpose is entertainment (vs accomplishing tasks)
3) Build specifications will vary depending on platform
4) Software has a story, mechanics, and aesthetic theme; hardware has justified use-case

Infrastructure

1) Intended for use by engineers (not the end-user)
2) Primary purpose is to facilitate the development of other engineering artifacts
3) Build specifications are ***universal***, regardless of target platform
4) Has an application programming interface (API) and/or physical interface

Build Specifications

The build specifications for projects vary depending on project type and, in some cases, target platform. These build specifications must be met for all project milestones. We will consider exceptions on a case-by-case basis.

1) Software releases should be prepared as a source distribution package **buildable and installable** by instructors via a single command (e.g., "`pip install project`") on Windows 10/11, Ubuntu 20.04 (native/WSL), Android 10.0, iOS, or MacOS Mojave. Teams must also provide a binary distribution.
2) Hardware releases should provide detailed schematics and other design documents **sufficient to completely understand** the device, along with a physical artifact.
3) Infrastructure projects must include at least one sample use case (application/device) to demonstrate functionality as proof of concept.

Team Coordination

The following expectations apply to each team and its members' work and coordination with one another.

Contributions

1) All students must contribute, in approximately equal proportion, to the technical aspects of the project.
2) Serving as a Team Lead does not absolve a team member from the expectation of technical contributions.
3) Instructors of the course may elect to make grade adjustments for unequal contributions.
4) Team members are obligated to complete peer review documentation honestly and with candor.
5) Teams must meet weekly and must keep a meeting log of attendance. This log should be in the repository. This meeting must be distinct from the client meeting.

Design Repository

1) Team must use both a repository system (e.g., GitHub) and issue tracking system (e.g., Trello). The issue tracking system must include the capability to assign tasks to members, and it must be utilized.
2) Repository commits should be made for all schematics, code, and resources developed, without exception. Git LFS support can be used for very large files. For physical artifact work, teams must maintain a detailed log of work accomplished and by whom. Logs must be committed regularly to the repository
3) Any contribution should be placed in the repository, including planning documents that are developed, in order to facilitate identification of time and effort spent.
4) Contributions should be committed to the repository by the individual contributing; items should be merged within the repository system directly to create a clear trail of effort and contribution.
5) All elements must be merged to "master" or "main" by project milestone deadlines.
6) Projects must have a "Contributions" file in the repository that outlines work on all non-logged tasks (which are therefore not reflected in the repository). This should include the name of the person who completed the task, date(s) of work, and a short description of the task itself.

Presentations

1) All students must contribute to the presentation directly – i.e., everyone must present.
2) All students must take part in the planning and preparation of the presentations to receive credit.
3) All team members should have approximately equal speaking time during the presentations.