UF UNIVERSITY of FLORIDA
Herbert Wertheim
College of Engineering

Performant Programming
CEN4930

M3: Production Release (Application)

Overview

As a team, you will complete and deliver a ***production release***, implementing all the features of your project, as your final deliverable. As the name implies, the production release must be a complete product with all final features integrated, functional, and polished. Implementation of all aspects of every feature must be completed, and user testing should have been completed. The production release is the final version of the application that is delivered to clients, and as such it should be usable by a typical user. The production release should represent approximately 100 hours of effort ***per person***, cumulatively (including time spent on the prototype).

Specification

The deliverable should meet the following requirements.

User Experience

The user feedback system, including UI and sensory response elements, should complete and polished in the production release. All elements of user control should be completed.

Interface

All user interface elements should be visible, and usable, and polished. Options should be intuitive and consistent. Additionally, the currently selected button(s), menu option(s), or other dialog element(s) should be highlighted. General status changes must be visible in the UI.

Navigation

All user mechanics should be functional and polished. Controls should be thoroughly tested. There should be no bugs. Controls should not require significant time for acclimation; they should be predictable and easy to use. Features should be easily discoverable and well-explained. They should not overwhelm the new users.

User Perception

Application responsive and use must be intuitive to users. Teams should have solicited user testing from non-team members to ensure general usability. Users should report an enjoyable experience. The release should help users attain stated goals with minimal frustration.

Responsiveness

Interactions with the user interface must be indicated by the sensory experience (visually and/or audibly). Changes in application state should be indicated on or near the relevant interface elements. Additionally, completion or failure of tasks must be clearly indicated to the user.

Build Quality

Projects receiving a perfect build score will have no detectable bugs. All content, including art and code systems, should be present and polished.

Robustness

There should be no crashes or glitches in the production release of the project.

Consistency

Except where explicitly by design, the application should act predictably – for the same input / use case, it should yield the same result. When / if behavior is unpredictable, it must be for a clear and compelling reason.

Aesthetic Rigor

No cosmetic issues should be present. Under no circumstances should aesthetic issues cause the application to be unusable. All assets (e.g., images and sounds) should be present and functional within the context of the project. Assets and their integration within the application should be polished.

Features

All features should be complete, with edge cases tested and accounted for, and with failure modes anticipated and handled.

Front-End

For each possible use-case, the control scheme(s) must be implemented, usable, and smooth within the UI of the application. State changes must be reflected in the UI. This must connect to the *persistent state*.

Persistent State

For each possible use case, the release should make proper use of a data store to update the state of the application. This must connect to the *front-end* and the *back-end*.

Back-End

For each possible use case, all data processing steps must be fully implemented. This must connect to the *persistent state*.

Submissions

Your submission will be a tgz (gzipped tar) archive including the following:

- Source code and assets for submission
- README file

Source & Assets

Submissions should include all source code, including build configurations and scripts (e.g., setup.py and/or Cargo.toml) and run scripts as necessary. The project should be runnable using an executable command without parameters and should be prepared by build scripts. For server applications, a command to initialize the server state for first run may be included. Submissions should also include all pre-built assets necessary for application function (e.g., pre-built database collections necessary for initial function and/or visual/audio assets).

README File

A README file, either in plaintext or markdown (UTF-8) should be included. The README should have, at a minimum, the following information:

- A description of the project
- One-line command for building and installation
- Link to the repository for the project (e.g., GitHub) [Note: if private, you must add instructors]
- For mobile projects, a link to the application package (APK)
- Run executable command and, if application, initialization command